



**TUGAS AKHIR - TE 141599**

**PERANCANGAN PERANGKAT ANTARMUKA BERBASIS  
PENGENALAN SUARA PADA PURWARUPA MESIN  
CETAK HURUF BRAILLE ITS**

Nicolas Rezadhi Pradipta  
NRP 07111440000065

Dosen Pembimbing  
Ir. Tasripan, M.T.  
Dr. Ir. Hendra Kusuma, M.Eng.Sc

DEPARTEMEN TEKNIK ELEKTRO  
Fakultas Teknologi Elektro  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018





**FINAL PROJECT - TE 141599**

***VOICE RECOGNITION-BASED INTERFACE HARDWARE  
DESIGN ON ITS BRAILLE EMOSSERS***

Nicolas Rezadhi Pradipta  
NRP 07111440000065

Supervisor  
Ir. Tasripan, M.T.  
Dr. Ir. Hendra Kusuma, M.Eng.Sc

DEPARTEMENT OF ELECTRICAL ENGINEERING  
Faculty of Electrical Technology  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018



## **PERNYATAAN KEASLIAN TUGAS AKHIR**

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “PERANCANGAN PERANGKAT ANTARMUKA BERBASIS PENGENALAN SUARA PADA PURWARUPA MESIN CETAK HURUF BRAILLE ITS ” adalah benar-benar hasil karya intelektual sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya orang lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Surabaya, 24 Juni 2018

Nicolas Rezadhi Pradipta  
NRP. 07111440000065

-----*Halaman ini sengaja dikosongkan*-----

**PERANCANGAN PERANGKAT ANTARMUKA BERBASIS  
PENGENALAN SUARA PADA PURWARUPA MESIN CETAK  
HURUF BRAILLE ITS**

**TUGAS AKHIR**

Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Teknik  
Pada  
Bidang Studi Elektronika  
Departemen Teknik Elektro  
Institut Teknologi Sepuluh Nopember

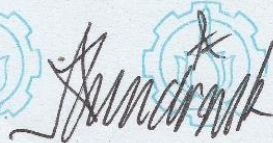
Menyetujui:

Dosen Pembimbing I

Dosen Pembimbing II



Ir. Tasripan, M.T.  
NIP. 196204181990031004



Dr. Ir. Hendra Kusuma, M. Eng.Sc  
NIP. 196409021989031003



-----*Halaman ini sengaja dikosongkan*-----



# **PERANCANGAN PERANGKAT ANTARMUKA BERBASIS PENGENALAN SUARA PADA PURWARUPA MESIN CETAK HURUF BRAILLE ITS**

**Nama : Nicolas Rezadhi Pradipta**  
**Pembimbing I : Ir. Tasripan, M.T.**  
**Pembimbing II : Dr. Ir. Hendra Kusuma, M. Eng.Sc**

## **ABSTRAK**

Pada pembuatan Purwarupa Mesin Cetak Huruf Braille ITS sejak tahun 2012 yang dikerjakan oleh Tim FTE ITS yang bekerjasama dengan Direktorat Pembinaan PK-LK Dikdas, Direktorat Jenderal Pendidikan Dasar, Kementerian Pendidikan dan Kebudayaan, sampai pada tahun 2017 sudah banyak mengalami perkembangan. Banyak nya perkembangan ini membuat purwarupa ini memiliki kelebihan dan banyak fitur. Salah satunya kemudahan dalam pengoperasian serta kompatibel dengan sistem operasi komputer modern. Dengan adanya sistem operasi komputer modern ini sangat memudahkan operator purwarupa Mesin Cetak Huruf Braille ITS. Tetapi sistem operasi ini masih memiliki kemelahan yaitu masih membutuhkan operartor non-tunanetra. Dengan adanya perangkat antarmuka berbasis suara yang terkoneksi pada sistem operasi, purwarupa ini dapat dioperasikan oleh operator penyandang tunanetra juga sehingga dapat lebih memudahkan pengoperasian purwarupa. Dirancang menggunakan *Voice Recognition Module by Elechouse* dan *STM32F4 Discovery*, sistem ini memasukkan database dari suara yang sudah dilatih berupa data hexadecimal kedalam STM32 yang lalu berkomunikasi dengan purwarupa. Dengan 4 perintah suara untuk mengoperasikan purwarupa ini, hasil yang didapat bekerja dengan rata-rata keberhasilan pengenalan suara sebesar 60,94% menggunakan referensi suara orang lain dan 77,92% menggunakan referensi suara sendiri.

**Kata Kunci :** Kontrol Suara, Mesin Cetak Braille, Modul Pengenalan Suara, Perangkat Antarmuka

-----*Halaman ini sengaja dikosongkan*-----

## **VOICE RECOGNITION-BASED INTERFACE HARDWARE DESIGN ON ITS BRAILLE EMOSSERS**

**Name** : Nicolas Rezadhi Pradipta

**Advisor 1<sup>st</sup>** : Ir. Tasripan, M.T.

**Advisor 2<sup>nd</sup>** : Dr. Ir. Hendra Kusuma, M. Eng.Sc

### **ABSTRACT**

*Since 2012, the manufacture of ITS Braille Embossers Prototype conducted by Faculty of Electrical Engineering in collaboration with Direktorat Pembinaan PK-LK Dikdas, Direktorat Jenderal Pendidikan Dasar, Kementrian Pendidikan dan Kebudayaan until 2017 has been a lot of progress and development. Many of these developments have made this prototype more advance and has many features. Among other thing is user friendly, easily controlled and maintenance, and compatible with modern computer operating system. With this modern computer operating system, it is very easy to operate this ITS Braille Embossers. But this operating system still need a non-visually impaired operator. With this hardware interface that based on voice recognition connected to the operating system, this prototype can be operated by visually impaired operator and also facilitate the operation of this prototype. Designed using Voice Recognition Module by Elechouse and STM32F4 Discovery, this system includes a database of trained voice in the form of hexadecimal data into STM32 which then communicates with the prototype. With 4 voice commands to control the prototype, the result works with average voice recognition rate 60,94% using other voices reference and 77,92% using own voice reference.*

**Keywords** : Speech Control, Braille Embossers, Speech Recognition Module, Hardware Interface

-----*Halaman ini sengaja dikosongkan*-----

## KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Tuhan Yang Maha Esa, atas terselesaikannya laporan Tugas Akhir yang berjudul “Perancangan Perangkat Antarmuka Berbasis Pengenalan Suara Pada Purwarupa Mesin Cetak Huruf Braille ITS untuk Operator Tunanetra”. Penulis sadar bahwa dengan tanpa adanya bantuan dari pihak lain, penulis tidak akan dapat menyelesaikan laporan tugas akhir ini. Dengan segala hormat dan rasa syukur yang sedalam-dalamnya penulis mengucapkan terima kasih kepada:

1. Tuhan Yang Maha Esa yang telah memberi berkat serta rahmat yang luar biasa sehingga penulis dapat menyelesaikan tugas akhir ini terlepas dari segala masalah yang ditemui.
2. Keluarga penulis, papa, Pius Benny Benjamin, mama, Yohana Avila Nawang Windarti, dan kedua adik saya Archangela Reza Serafine Anindita Putri, Michelle Reza Zefanya Elena Putri yang tanpa henti memberikan dukungan moral dan semangat.
3. Dosen Pembimbing 1, Ir. Tasripan, M.T. atas segala bimbingan, motivasi, ilmu dan arahan pada penulis sehingga dapat terselesaikannya tugas akhir ini.
4. Dosen Pembimbing 2, Dr. Ir. Hendra Kusuma, M.Eng.Sc. atas segala bimbingan, motivasi, ilmu dan arahan pada penulis sehingga dapat terselesaikannya tugas akhir ini.
5. Regita Andrienne, yang selalu mendampingi penulis menjalani semester akhir dan seterusnya sampai terselesaikan nya tugas akhir ini.
6. Mentor-mentor saya, Kamal Arief dan Anggarjuna Puncak S.T. yang memberi ilmu tambah dan solusi-solusi atas masalah yang saya temui selama mengerjakan tugas akhir ini.
7. Seluruh teman-teman bidang studi elektronika 2014 dan 2015
8. Teman-teman E54 dan adik-adik E55 dan E56.

Terima kasih juga kepada semua pihak yang mungkin tidak dapat penulis sebutkan satu persatu. Penulis juga sadar bahwa banyaknya kekurangan dari penulis dalam penulisan laporan ini. Penulis dengan terbuka menerima masukan, kritik, dan saran agar laporan ini dapat bermanfaat kedepannya.

**“Jangan berhenti saat lelah, Berhentilah saat sudah selesai”**

Surabaya, **JUNI 2018**

Penulis

-----*Halaman ini sengaja dikosongkan*-----

# DAFTAR ISI

ABSTRAK .....	i
ABSTRACT .....	iii
KATA PENGANTAR.....	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR .....	ix
DAFTAR TABEL .....	xi
 BAB I PENDAHULUAN .....	 1
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah .....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan .....	2
1.5 Metodologi Penelitian .....	3
1.6 Sistematika Laporan.....	4
1.7 Relevansi .....	4
 BAB II LANDASAN TEORI .....	 7
2.1 Produksi Sinyal Suara Manusia Secara Fisiologis .....	7
2.2 Pengenalan Suara .....	8
2.3 Voice Recognition Module V3 .....	10
2.3.1 Mikrofon .....	14
2.4 Mikrokontroler STM32F4 <i>Discovery</i> .....	15
2.4.1 UART .....	17
2.5 Purwarupa Mesin Cetak Huruf Braille ITS .....	18
 BAB III PERANCANGAN PERANGKAT ANTARMUKA .....	 21
3.1 Diagram Blok Sistem Antarmuka .....	21
3.2 Perancangan Antarmuka Otomatis .....	22
3.2.1 Proses Pelatihan <i>Voice Recognition 3 Module</i> .....	22
3.2.2 Proses Pengambilan Data Serial .....	27
3.2.3 Flowchart Program Antarmuka Otomatis.....	29
3.3 Perancangan Antarmuka Manual .....	33
3.4 Perancangan <i>Hardware</i> .....	34
3.4.2 Perancangan Perangkat Elektrik.....	34
 BAB IV PENGUJIAN DAN ANALISA PERANGKAT ANTARMUKA.....	 36

4.1 Pengujian Supply Tegangan .....	37
4.2 Pengujian Pengenalan Suara pada Antarmuka Otomatis .....	38
4.2.1 Uji Coba Pria dengan Referensi Database Admin Pria ....	39
4.2.2 Uji Coba Wanita dengan Referensi Database Admin Wanita .....	43
4.2.3 Uji Coba dengan Referensi Sendiri .....	46
4.3 Pengujian Komunikasi antara Perangkat Antarmuka dengan Purwarupa Mesin Cetak Braille ITS .....	50
5BAB V PENUTUP .....	57
5.1 Kesimpulan .....	57
5.2 Saran .....	57
DAFTAR PUSTAKA .....	59
LAMPIRAN .....	61
BIODATA PENULIS .....	73



## DAFTAR GAMBAR

Gambar 2.1 Komponen Fisiologis Organ Suara Manusia [4] .....	8
Gambar 2.2 Teknik Pengenalan Suara[5] .....	9
Gambar 2.3 Blok Diagram Pengenalan Suara Fase Latihan dan Fase Pencobaan[6].....	9
Gambar 2.4 Bentuk Fisik VR3 <i>Module</i> [7] .....	11
Gambar 2.5 Daftar Perintah Protokol pada Arduino .....	12
Gambar 2.6 Protokol untuk Melatih Suara[7] .....	12
Gambar 2.7 Protokol untuk Melatih Suara dengan Penanda[7] .....	13
Gambar 2.8 Protokol untuk Memasukan Index Suara pada <i>Recognizer</i> [7] .....	13
Gambar 2.9 Protokol untuk Menghapus Index Suara pada <i>Recognizer</i> [7] .....	14
Gambar 2.10 Rangkaian Mikrofon Elektret .....	14
Gambar 2.11 Bentuk Fisik Papan Tunggal STM32F4-Discovery[8]...	15
Gambar 2.12 Top Layout[8] .....	16
Gambar 2.13 Bottom Layout[8].....	16
Gambar 2.14 Pinout STM32F4 Discovery[9] .....	17
Gambar 2.15 Purwarupa Mesin Cetak Braille ITS.....	18
Gambar 2.16 Proses Komunikasi pada Purwarupa .....	19
Gambar 3.1 Diagram Blok Perangkat Sistem Antarmuka.....	21
Gambar 3.2 Koneksi Arduino dengan <i>Voice Recognition Module V3</i> .	23
Gambar 3.3 <i>vr sample train.ino</i> Untuk Melatih Modul .....	23
Gambar 3.4 Daftar <i>command</i> Untuk Melatih Modul .....	24
Gambar 3.5 Proses Pelatihan Kata .....	26
Gambar 3.6 Hasil Pengenalan Suara pada Serial Monitor .....	27
Gambar 3.7 Koneksi USB FTDI dengan <i>Voice Recognition</i> <i>Module V3</i> .....	27
Gambar 3.8 Data hex berdasarkan pengenalan pada <i>Hterm</i> .....	28
Gambar 3.9 <i>Flowchart Auto Interface</i> .....	30
Gambar 3.10 Gambaran Desain <i>Hardware</i> .....	34
Gambar 3.11 Wiring Diagram Perangkat Antarmuka.....	35
Gambar 4.1 Supply Tegangan STM32F4 .....	36
Gambar 4.2 Supply Tegangan pada <i>VR3 Module</i> .....	37
Gambar 4.3 Diagram Batang Keberhasilan Rekognisi Perintah Suara pada Pria .....	40

Gambar 4.4 Diagram Lingkaran Persentase Keberhasilan Rekognisi Perintah Suara oleh <i>Voice Recognition Module V3</i> pada pria ..	41
Gambar 4.5 Diagram Batang Keberhasilan Rekognisi Perintah Suara pada Wanita .....	44
Gambar 4.6 Diagram Lingkaran Persentase Keberhasilan Rekognisi Perintah Suara oleh <i>Voice Recognition Module V3</i> pada wanita .....	44
Gambar 4.7 Diagram Batang Perbandingan Persentase Keberhasilan Rekognisi.....	47
Gambar 4.8 Pola Test Print 1 .....	49
Gambar 4.9 Pola Test Print 2.....	50
Gambar 4.10 Pola Test Print 3 .....	51
Gambar 4.11 Hasil Test Print 1 .....	52
Gambar 4.12 Hasil Test Print 2 .....	53
Gambar 4.13 Hasil Test Print 3 .....	54

## DAFTAR TABEL

Tabel 2.1 Parameter VR3 <i>Module</i> .....	11
Tabel 2.2 Konfigurasi Perintah Purwarupa .....	19
Tabel 3.1 Daftar Kata yang Dilatih .....	25
Tabel 3.2 Data Hex yang Dikirim Modul Berdasarkan Pengenalan suara Admin Pria .....	29
Tabel 3.3 Konfigurasi perintah dari STM32F4 ke purwarupa .....	33
Tabel 4.1 Rata-Rata Persentase Keberhasilan .....	38
Tabel 4.2 Jumlah kata “Test Print” yang Berhasil Terekognisi dari 10 kali pengucapan .....	38
Tabel 4.3 Jumlah kata “Satu” yang Berhasil Terekognisi dari 10 kali pengucapan .....	39
Tabel 4.4 Jumlah kata “Dua” yang Berhasil Terekognisi dari 10 kali pengucapan .....	39
Tabel 4.5 Jumlah kata “Tiga” yang Berhasil Terekognisi dari 10 kali pengucapan .....	40
Tabel 4.6 Rata-Rata Persentase Keberhasilan Pria dengan Referensi Admin .....	41
Tabel 4.7 Jumlah kata “Test Print” yang Berhasil Terekognisi dari 10 kali pengucapan .....	42
Tabel 4.8 Jumlah kata “Satu” yang Berhasil Terekognisi dari 10 kali pengucapan .....	42
Tabel 4.9 Jumlah kata “Dua” yang Berhasil Terekognisi dari 10 kali pengucapan .....	43
Tabel 4.10 Jumlah kata “Tiga” yang Berhasil Terekognisi dari 10 kali pengucapan .....	43
Tabel 4.11 Rata-Rata Persentase Keberhasilan Wanita dengan Referensi Admin .....	45
Tabel 4.12 Jumlah kata “Test Print” yang Berhasil Terekognisi dari 10 kali pengucapan .....	45
Tabel 4.13 Jumlah kata “Satu” yang Berhasil Terekognisi dari 10 kali pengucapan .....	46
Tabel 4.14 Jumlah kata “Dua” yang Berhasil Terekognisi dari 10 kali pengucapan .....	46
Tabel 4.15 Jumlah kata “Tiga” yang Berhasil Terekognisi dari 10 kali pengucapan .....	47

Tabel 4.16 Rata-Rata Persentase Keberhasilan Wanita dengan  
    Referensi Admin..... 48

Tabel 4.17 Hasil Komunikasi pada Purwarupa..... 51

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Sejak tahun 2012 lalu ITS sedang menggarap riset purwarupa dari Mesin Cetak Huruf Braille. Purwarupa Mesin Cetak Huruf Braille ini merupakan pengembangan dari mesin cetak dari norwegia yang dimiliki SLB YPAB. Dengan bekerja sama dengan Direktorat Pembinaan PK-LK Dikdas, Direktorat Jenderal Pendidikan Dasar, dan Kementerian Pendidikan dan Kebudayaan, Tim dari FTE ITS membuat purwarupa dari Mesin Cetak Huruf Braille dengan hampir 85% menggunakan suku cadang lokal buatan Indonesia. Purwarupa yang dikembangkan oleh Tim FTE ITS ini memiliki kelebihan dan beberapa fitur antara lain adalah hemat energi, mudahnya pengoperasian dan pemeliharaannya, mampu mencetak karakter normal braille sebanyak 400 karakter/detik dengan maksimum 42 karakter/baris, dan dapat mencetak langsung pada 2 sisi kertas[1]. Purwarupa Mesin Cetak Huruf Braille ITS ini mempunyai manfaat yang besar bagi para penyandang tunanetra, terutama sebagai media untuk memudahkan pembelejaraan pada Sekolah Luar Biasa yang ada di Indonesia.

Dalam pengoperasian sebuah alat diperlukan adanya sistem antarmuka (*interface*) yang dapat memudahkan interaksi antara manusia sebagai *user* dengan alat yang akan dioperasikan. Pengembangan-pengembangan mengenai bagaimana interaksi antara manusia dan mesin terus melaju pesat, sektor pengembangan ini lebih sering dikenal sebagai *Human Machine Interface* atau *Human Machine Interaction* atau *Human Computer Interaction*. Sejak awal penggarapan riset Purwarupa Mesin Cetak Huruf Braille ITS ini sudah terdapat banyak fitur-fitur yang dikembangkan untuk membuat purwarupa ini menjadi semakin sempurna, namun masih ada beberapa kelemahan yang sebenarnya dapat dikembangkan agar lebih memudahkan pengoperasiannya. Salah satu kelemahannya yaitu dalam hal pengoperasian. Walaupun sistem operasinya sudah berbasis komputer modern, purwarupa Mesin Cetak Huruf Braille ITS ini masih membutuhkan operator yang bukan merupakan penyandang tunanetra. Jika operator nya juga merupakan penyandang tunanetra, maka harus adanya upaya dari mereka untuk menghafalkan dan juga membiasakan/latihan untuk mengoperasikan dengan jangka

waktu yang cukup lama, untuk itu dalam memudahkan penyandang tunanetra untuk dapat mengoperasikan purwarupa Mesin Cetak Huruf Braille ITS ini dibutuhkan perangkat antarmuka yang dibuat khusus untuk penyandang tunanetra.

Dengan adanya perangkat antarmuka dengan basis pengenalan suara, ditambah lagi dengan fitur tombol yang bersuara para penyandang tunanetra dapat dengan mudah mengoperasikannya tanpa harus menghafal tombol operasi dari purwarupa mesin itu sendiri.

## **1.2 Perumusan Masalah**

Permasalahan yang akan dibahas dalam tugas akhir ini:

1. Proses pelatihan perintah suara pada modul *Voice Recognition Module V3* melalui papan tunggal STM32F4 *Discovery*
2. Perancangan tombol *keypad* dengan *output* suara
3. Pengiriman data yang sesuai dengan perintah dari perangkat antarmuka ke purwarupa Mesin Cetak Huruf Braille ITS

## **1.3 Batasan Masalah**

Batasan masalah pada tugas akhir ini adalah:

1. Sistem dilatih dengan intonasi berbicara dan warna suara pria dan wanita
2. Data masukan suara hanya dengan bahasa Indonesia dengan jumlah kata terbatas 5 kata yaitu “test print”, “satu”, “dua”, “tiga”

## **1.4 Tujuan**

Tujuan yang ingin dicapai dalam tugas akhir ini:

1. Menghasilkan sistem pengenalan perintah suara dengan papan tunggal STM32F4 *Discovery*
2. Membuat rancangan tombol eksekusi dengan *output* suara
3. Mengirim data yang sesuai dengan perintah dari perangkat antarmuka ke purwarupa Mesin Cetak Braille ITS

## 1.5 Metodologi Penelitian

Dalam penyelesaian tugas akhir ini metode yang dikerjakan adalah sebagai berikut:

1. Studi Literatur  
Pada tahap ini dilakukan pengumpulan dasar-dasar teori dan landasan-landasan yang mendukung untuk dijadikan acuan pada pengerjaan tugas akhir. Dasar teori dapat diambil dari buku, skripsi, tesis, jurnal, artikel, dan video tutorial yang dapat diakses melalui internet.
2. Perancangan Sistem  
Pada tahap ini dilakukan perancangan sistem yang terbagi dua menjadi sistem antarmuka otomatis dan sistem antarmuka manual. Pada perancangan sistem antarmuka otomatis digunakan *Elechouse Voice Recognition module V3* sebagai modul pengenalan suara, *USB to TTL*, papan tunggal *STM32F4 Discovery* sebagai microcontroller. Pada perancangan sistem antarmuka manual digunakan *push button* dan *buzzer*, papan tunggal *STM32F4 Discovery* sebagai mikrocontroller.
3. Perancangan *Software*  
Pada tahap ini *software* dirancang berdasarkan skema alur dari sistem yang sudah dirancang. Perancangan *Software* dilakukan menggunakan *Keil uVision 5*, dan *STM32Cube MX* untuk mengkonfigurasi papan tunggal *STM32F4 Discovery*.
4. Perancangan *Hardware*  
Pada tahap ini *Hardware* dirancang untuk membungkus sistem dan *software* yang sudah terancang agar sistem ini dapat bekerja secara fungsional dan terlihat *compact*.
5. Pengujian Sistem  
Pada tahap ini dilakukan pengujian terhadap sistem antarmuka otomatis yaitu pengujian pengenalan suara dengan menggunakan 10 operator pria dan 10 operator wanita lalu pengujian komunikasi serial dari mikrokontroler ke purwarupa Mesin Cetak Huruf Braille ITS.
6. Penulisan Laporan Tugas Akhir  
Merupakan tahap akhir dari proses pengerjaan tugas akhir. Isi dari laporan tugas akhir berkaitan dengan tugas akhir yang dikerjakan meliputi pendahuluan, teori dasar, perancangan alat, pengujian dan penutup.

## **1.6 Sistematika Laporan**

Pembahasan tugas akhir ini akan disusun dengan sistematika sebagai berikut:

### **BAB I PENDAHULUAN**

Bab pendahuluan, membahas mengenai latar belakang pemilihan topik, perumusan masalah dan batasannya, tujuan penelitian, metodologi, sistematika laporan, dan relevansi dari penelitian yang dilakukan.

### **BAB II LANDASAN TEORI**

Bab teori penunjang merupakan bab yang berisi dasar teori dan literatur yang digunakan sebagai acuan dalam proses pengerjaan tugas akhir ini.

### **BAB III PERANCANGAN PERANGKAT ANTARMUKA**

Bab perancangan sistem berisi realisasi dari sistem antarmuka yang terdiri dari perangkat lunak, perangkat keras, sistem antarmuka dan penggabungan dari seluruh sistem.

### **BAB IV PENGUJIAN DAN ANALISA PERANGKAT ANTARMUKA**

Bab pengujian dan analisa adalah hasil pengujian dari sistem yang telah dibuat dengan beberapa analisa yang berkaitan dengan pengujian

### **BAB V PENUTUP**

Bab penutup berisi kesimpulan yang diperoleh dari pembuatan tugas akhir, dan juga saran-saran untuk pengembangan lebih lanjut.

## **1.7 Relevansi**

Hasil yang diperoleh dari tugas akhir ini diharapkan dapat bermanfaat untuk referensi pembuatan *interface* untuk alat-alat yang membutuhkan *user* penyandang tunanetra terutama untuk pembuatan



purwarupa Mesin Cetak Huruf Braille ITS sehingga dapat membantu penyandang tunanetra dalam berinteraksi dengan teknologi.

-----*Halaman ini sengaja dikosongkan*-----

## **BAB II**

### **LANDASAN TEORI**

Dalam bab ini akan dibahas mengenai teori yang menunjang sistem pengenalan suara sampai kepada proses mikrokontroller lalu. Pemahaman mengenai proses mekanisme terproduksinya suara manusia menjadi sinyal suara, filter untuk sinyal suara, dan proses sampling serta kuantisasi sinyal suara untuk didigitalisasikan sampai masuk kedalam mikrokontroller serta sistem kerja *voice recognition*. Juga terdapat beberapa komponen yang diperlukan untuk membuat sistem ini seperti *voice recognition module V3* sebagai module untuk melatih pengenalan suara, papan tunggal STM32F4 *Discovery* sebagai mikrokontroller yang memproses algoritma sistem antarmuka.

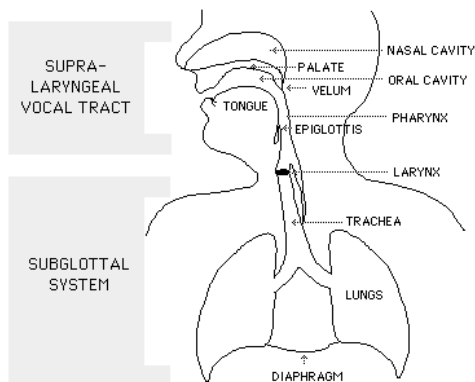
#### **2.1 Produksi Sinyal Suara Manusia Secara Fisiologis**

Suara adalah salah satu elemen yang berperan penting dalam kehidupan manusia karena suara adalah salah satu media manusia untuk berkomunikasi. Suara juga dapat menyatakan informasi pribadi seperti status sosial, kepribadian seseorang, dan juga kondisi emosional dari pembicaranya[2]. Suara merupakan alat utama dalam hal berkomunikasi baik antar manusia maupun manusia dengan mesin. Karena itu sebelum kita bisa memproses sinyal suara dalam bentuk sinyal digital, kita harus terlebih dahulu mengerti bagaimana proses sinyal suara terbentuk dari dalam tubuh manusia.

Secara fisiologis, suara manusia diproduksi melalui 3 proses utama yaitu proses pernafasan yang mana dikerjakan oleh paru-paru sebagai sumber energi. Proses kedua adalah proses pembunyian yang dikerjakan oleh laring. Setelah disalurkan dari paru-paru melalui trakea, energi berupa aliran udara akan sampai pada laring yang bekerja sebagai resonator. Proses pada laring yang paling utama adalah pita suara. Pita suara yang terbuka dan tertutup secara cepat membuat aliran udara dari trakea menjadi terhalang oleh pita suara dan membuat pita suara bergetar dan menghasilkan suara.

Kedua organ (baca: paru-paru dan laring) bersama menyesuaikan nada, kenyaringan, dan kualitas suara, dan selanjutnya menghasilkan pola prosodi suara[3]. Proses ketiga adalah artikulasi, suara yang dihasilkan oleh pita suara di modifikasi menjadi suara yang dapat dimengerti dan menghasilkan vokal dan konsonan. Artikulasi ini lah yang membuat

adanya warna suara pada masing-masing manusia. Pada gambar 2.1 dapat kita lihat terbaginya komponen fisiologis dari organ suara manusia yaitu, komponen subglotal, laring, dan komponen supraglotal (supralaryngeal). Ketiga komponen inilah yang membedakan tempat terjadinya 3 proses yang sudah dijelaskan diatas.

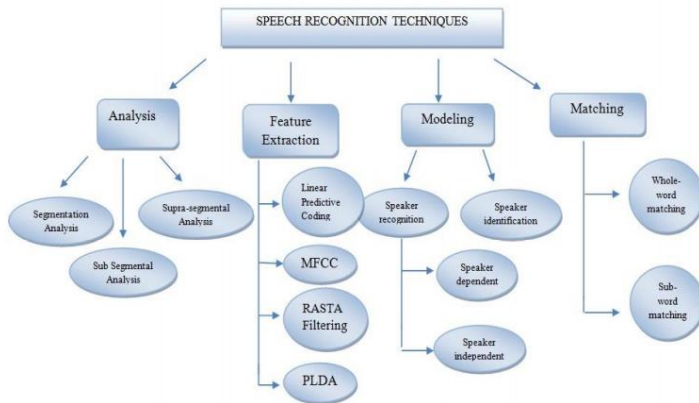


Gambar 2.1 Komponen Fisiologis Organ Suara Manusia [4]

## 2.2 Pengenalan Suara

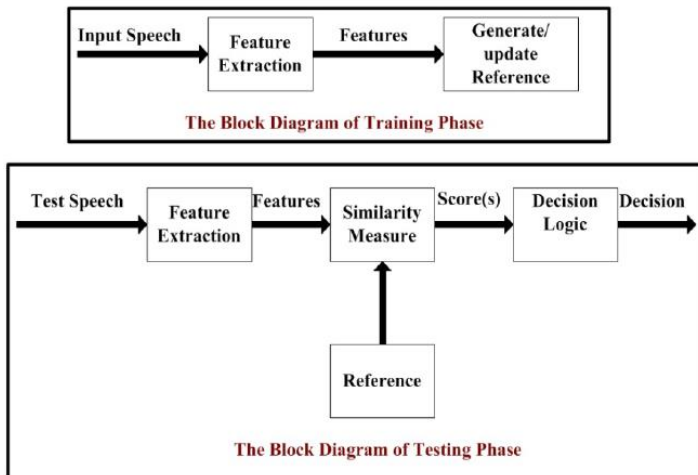
Pengenalan suara merupakan salah satu metode dari *Human-Machine Interaction* yang dikembangkan agar manusia dapat mengoperasikan suatu mesin dengan menggunakan suara. Proses pengenalan suara dapat dibagi menjadi pengenalan pembicara dan pengenalan ucapan. Pada pengenalan pembicara (*Speaker Recognition*) pengidentifikasian suara nya berdasarkan suara orang yang berbicara sedangkan pada pengenalan ucapan (*Speech Recognition*) yang di identifikasi adalah suara berdasarkan kata yang diucapkan. Sistem pengenalan suara memiliki tujuan utama untuk mengerti lalu memberi aksi pada informasi suara yang diberikan[5]. Seperti yang dapat kita lihat pada Gambar 2.2 adalah teknik dari pengenalan suara.

Pada umumnya proses dari pengenalan suara dapat kita lihat pada Gambar 2.3. Dapat kita lihat pada Gambar 2.3 terdapat dua fase yaitu fase pelatihan dan fase percobaan atau pengenalan. Pada fase pertama yaitu fase pelatihan, suara masuk kedalam sistem ekstraksi fitur dan disimpan dalam suatu perangkat. Suara-suara yang terkumpulkan dan disimpan dalam perangkat dianggap sebagai referensi.



Gambar 2.2 Teknik Pengenalan Suara[5]

Proses pelatihan ini biasanya dilakukan beberapa kali, karena merupakan proses *neural network* proses latihan ini jika dilakukan lebih banyak akan membuat suara yang akan dikenali lebih akurat.



Gambar 2.3 Blok Diagram Pengenalan Suara Fase Latihan dan Fase Pencobaan[6]

Pada fase percobaan atau pengenalan, dengan berbekal data referensi pada perangkat, sinyal suara dimasukkan lagi kedalam sistem ekstraksi fitur lalu sinyal suara akan di bandingkan dengan data referensi, kemudian diterjemahkan sebagai data keluaran. Data keluaran dapat berupa tulisan ataupun perintah.

Pada Gambar 2.3 dapat kita lihat adanya *feature extraction* atau ekstraksi fitur. Pada ekstraksi fitur inilah terdapat banyak metode yang dikembangkan untuk membuat sistem pengenalan suara. Ekstraksi fitur merupakan bagian utama pada sistem pengenalan suara. Hal ini dapat dianggap sebagai jantung dari sistem[5]. Terdapat 4 ekstraksi fitur yang terkenal pada pengenalan suara yaitu *Linear Predictive Coding*, *Mel-Frequency Cepstrum*, *RASTA Filtering*, *Probabilistic Linear Discriminate Analysis*. Pada modeling sistem atau pengukuran kesamaan untuk membandingkan pola dari suara masukan dengan referensi biasanya digunakan *Hidden Markov Model* walaupun masih ada banyak model lain yang bisa digunakan.

### 2.3 Voice Recognition Module V3

Sistem *auto interface* yang memiliki fitur pengenalan suara pada tugas akhir ini menggunakan alat bantu berupa modul pengenalan suara yang diciptakan oleh Elechouse yaitu *Voice Recognition Module V3*. Ini merupakan modul pengenalan suara versi paling terakhir yang dikembangkan oleh perusahaan Elechouse. Modul pengenalan suara ini berbasis chip serial dengan memory flash winbond W25Q16DV. Dengan chip ini, modul dapat menyimpan 80 index perintah suara yang dapat di latih. Tetapi hanya 7 index perintah suara yang dapat bekerja dalam satu waktu. Perintah suara juga dapat dikelompokkan menjadi grup-grup yang berisi maksimal 7 perintah suara.

Terdapat dua cara dalam mengoperasikan modul ini yaitu port serial dan *general input pins*. Dan juga memiliki *general output pins* yang dapat diatur untuk menciptakan beberapa macam gelombang yang sesuai dengan perintah suara yang terekognisi. Modul ini memiliki fitur *Arduino library*, *user-control General Pin Output*, dan lebar kata yang dapat dilatih sebanyak 1500ms.

Parameter yang dimiliki modul ini dapat dilihat pada tabel 2.1. Bentuk fisik dapat dilihat pada gambar 2.8

<i>Voltage</i>	4.5 – 5.5 Volt
<i>Current</i>	<40mA

<i>Digital Interface</i>	<i>5V TTL for UART interface and GPIO</i>
<i>Analog Interface</i>	<i>3.5mm mono-channel microphone connector</i>
<i>Size</i>	<i>31mm X 50mm</i>
<i>Recognition Accuracy</i>	<i>99% (under ideal environment)</i>

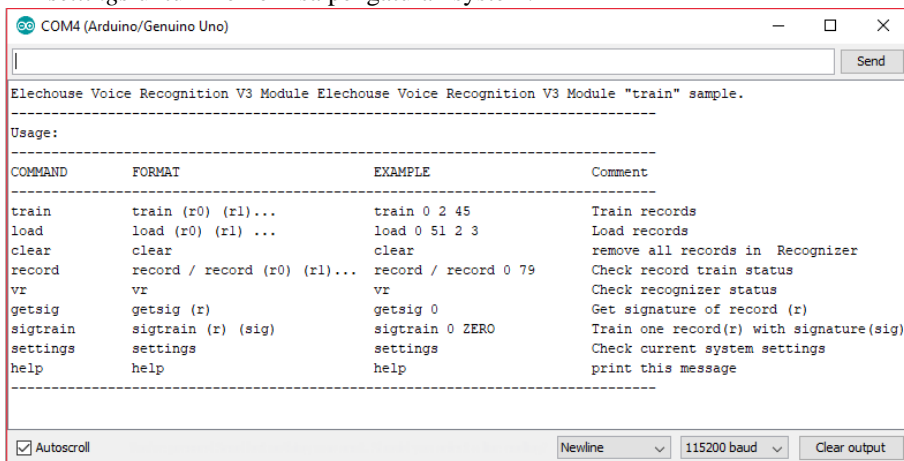
Tabel 2.1 Parameter VR3 Module



Gambar 2.4 Bentuk Fisik VR3 Module[7]

Untuk pengguna dapat mengoperasikan modul dibutuhkan komunikasi serial. Modul ini menyediakan 2 cara untuk komunikasi serial yaitu melalui protokol dengan komunikasi berupa hexadecimal dan juga menggunakan *library* Arduino yang memudahkan bahasa protokol menjadi bahasa *ascii*. Pengoperasian modul ini secara umum adalah melatih suara pada suatu index, membersihkan index dari *recognizer* dan memasukkan index ke *recognizer* tetapi masih ada beberapa fitur lainnya hanya saja digunakan dalam pengerjaan tugas akhir ini. Untuk melatih suara, pengguna perlu menentukan index yang akan dipakai untuk menyimpan hasil latihan suara tersebut. Seperti yang sudah dikatakan diatas, terdapat 80 index yang tersedia untuk menyimpan hasil latihan suara, terdapat juga fitur *signature* yang dapat digunakan untuk memberi penanda. Pada Arduino sangat mudah karena terdapat daftar untuk melakukan latihan suara, memasukan index pada *recognizer* dan membersihkan index pada *recognizer* melalui *library* yang diberikan seperti dapat dilihat pada Gambar 2.5

Terdapat perintah *train* untuk melatih kata pada index tertentu. *Load* untuk memasukkan index yang sudah dilatih pada *recognizer*. *Clear* untuk membersihkan *recognizer*, *vr* untuk memeriksa status *recognizer*, *getsig* untuk memeriksa kata penanda pada suatu index yang sudah dilatih, *sigtrain* untuk melatih kata pada index tertentu menggunakan penanda, *settings* untuk memeriksa pengaturan system.



Gambar 2.5 Daftar Perintah Protokol pada Arduino

Sedangkan jika menggunakan protokol hexadecimal dibutuhkan alat tambahan berupa USB Serial untuk mengkoneksikan modul dengan aplikasi terminal pada komputer. Dapat dilihat pada Gambar 2.6, 2.7, 2.8, 2.9 beberapa perintah protokol untuk mengoperasikan modul.

#### Train One Record or Records (20)

Train records, can train several records one time.

##### Format:

| AA | 03+n | 20 | R0 | ... | Rn | 0A |

##### Return:

| AA | LEN | 0A | RECORD | PROMPT | 0A |

| AA | 05+2\*n | 20 | N | R0 | STA0 | ... | Rn | STAn | SIG | 0A |

	Description
<b>R0-Rn</b>	Voice command record
<b>STA</b>	train result 0 -- Success 1 -- Timeout 2 -- Record value out of range
<b>n</b>	Number of trained voice command record

Gambar 2.6 Protokol untuk Melatih Suara[7]



### Train One Record and Set Signature (21)

Train one record and set a signature for it, one record one time.

**Format:**

| AA| 03+SIGLEN | 21 | RECORD | SIG | 0A | (Set signature)

**Return:**

| AA| LEN | 0A | RECORD | PROMPT | 0A | (train prompt)

| AA| 05+SIGLEN | 21 | N | RECORD | STA | SIG | 0A |

	Description
<b>RECORD</b>	Voice command record index
<b>SIG</b>	Signature string
<b>PROMPT</b>	Prompt string: <ul style="list-style-type: none"><li>• Speak now</li><li>• Speak again</li><li>• Success</li></ul>
<b>N</b>	Number of successful training voice commands

Gambar 2.7 Protokol untuk Melatih Suara dengan Penanda[7]

### Load a Voice Record or Records to Recognizer (30)

Load records(1~7) to recognizer of VR3, after execution the VR3 starts to recognize immediately.

**Format:**

| AA| 3+n | 30 | R0 | ... | Rn | 0A |

**Return:**

| AA| 3+2n | 30 | N | R0 | STA0 | ... | Rn | STAn | 0A |

	Description
<b>R0~Rn</b>	Voice Record index
<b>STA0~STAn</b>	Load result <ul style="list-style-type: none"><li>• <b>00</b> -- Success</li><li>• <b>FF</b> -- Record value out of range</li><li>• <b>FE</b> -- Record untrained</li><li>• <b>FD</b> -- Recognizer full</li><li>• <b>FC</b> -- Record already in recognizer</li></ul>

<b>N</b>	Number of successful training voice commands
----------	--

Gambar 2.8 Protokol untuk Memasukan Index Suara pada *Recognizer*[7]

### Clear Recognizer (31)

Stop recognizing, and empty recognizer of Voice Recognition Module. **Format:**

| AA | 02 | 31 | 0A |

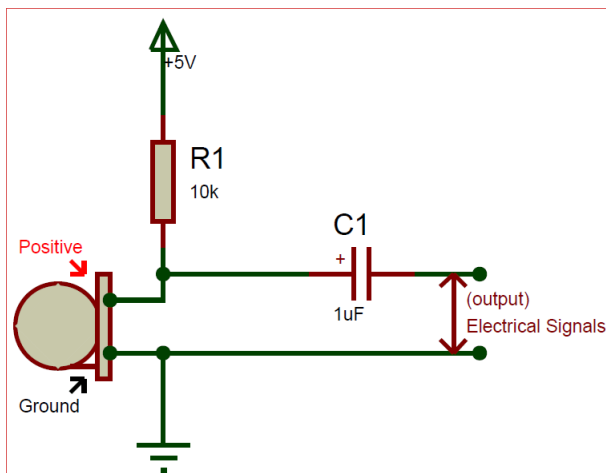
**Return:**

| AA | 03 | 31 | 00 | 0A |

Gambar 2.9 Protokol untuk Menghapus Index Suara pada  
*Recognizer*[7]

#### 2.3.1 Mikrofon

Pada modul *Voice Recognition Module V3* terdapat juga mikrofon bawaan, mikrofon ini merupakan mikrofon electret condenser. Mikrofon jenis ini sering ditemui pada perangkat-perangkat elektronik sehari-hari yang membutuhkan fungsi komunikasi dan perekaman audio. Hampir semua *smartphone*, *headset*, *earphone*, *desktop computer* menggunakan jenis mikrofon ini. Electret mikrofon merupakan tipe dari condenser mikrofon yang menggunakan prinsip kapasitor yaitu 2 membran tipis yang saling berdekatan, ketika diberikan gelombang suara jarak antar kedua membran akan berubah dan menghasilkan perbedaan kapasitansi yang dapat dikonversi menjadi sinyal elektrik.



Gambar 2.10 Rangkaian Mikrofon Elektret

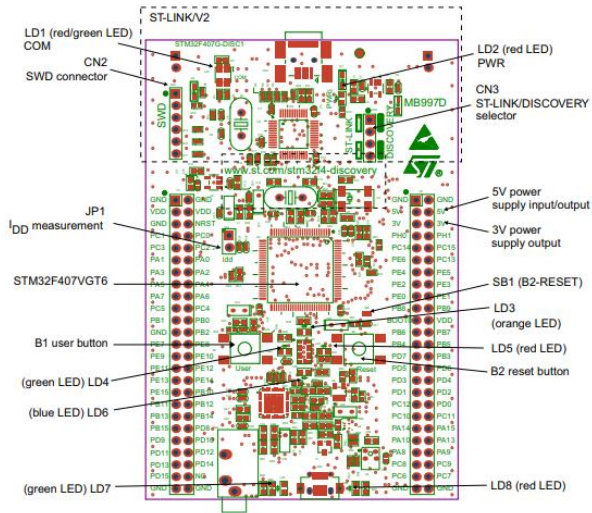
Elektret mikrofon hanya memiliki dua kaki dan butuh di pull up, sinyal elektrik sangat kecil maka dari itu electret mikrofon biasanya membutuhkan pre-amplifier sebelum masuk ke ADC.

## 2.4 Mikrokontroler STM32F4 *Discovery*

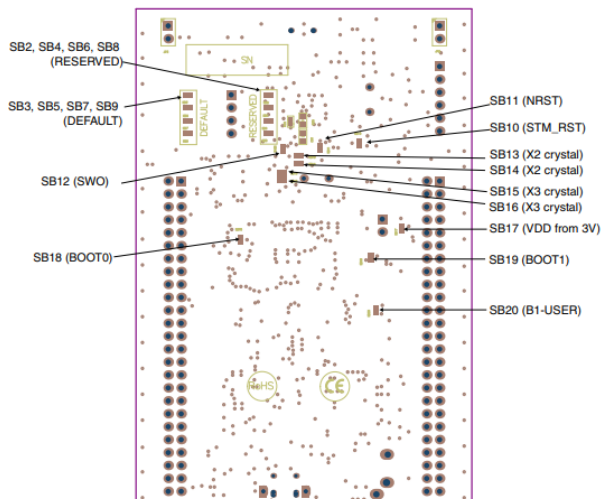
Papan tunggal STM32F4 *Discovery* adalah mikrokontroller menggunakan chip STM32F407VG yang merupakan mikrokontroller yang *high performance* dengan ARM Cortex-M4 32-bit yang dapat beroperasi hingga frekuensi 168 MHz. Papan tunggal ini dilengkapi dengan digital accelerometer, digital microphone, audio DAC yang terintegrasi dengan class D speaker driver, USB OTG micro-AB connector. Papan tunggal ini sangat cocok untuk aplikasi yang menggunakan audio. Pada gambar merupakan bentuk fisik dari papan tunggal ini



Gambar 2.11 Bentuk Fisik Papan Tunggal STM32F4-Discovery[8]



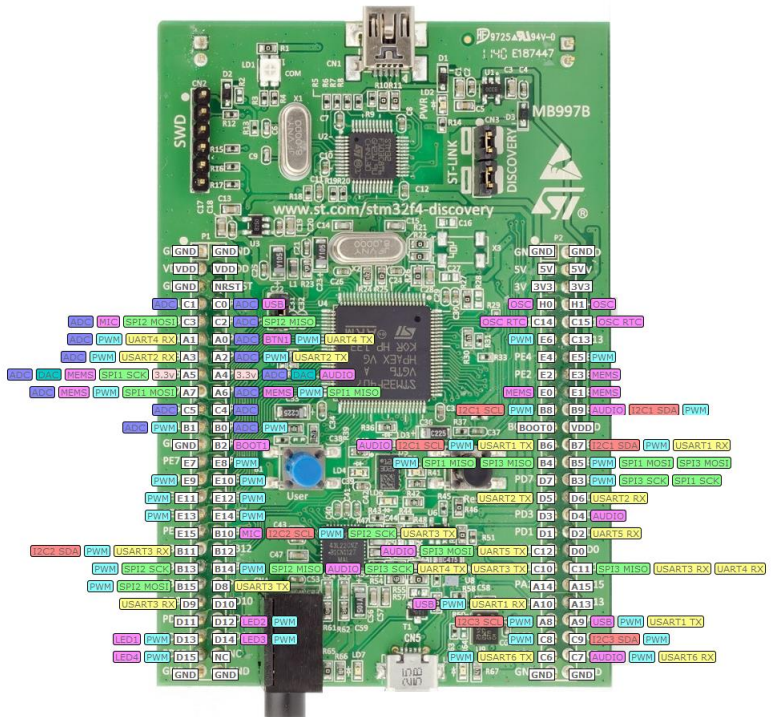
Gambar 2.12 Top Layout[8]



Gambar 2.13 Bottom Layout[8]

## 2.4.1 UART

UART( *Universal Asynchronous Receiver Transmitter*) merupakan salah satu bentuk komunikasi serial yang terdapat pada papan tunggal STM32F4 Discovery. Tugas utama dari UART adalah mengirim dan menerima data serial, konfigurasi pin dari UART hanya pin Rx dan pin Tx. Pin Rx merupakan pin *receiver* yaitu yang menerima data, dan pin Tx merupakan pin *transmitter* yaitu yang mengirim data. Papan tunggal STM32F4 Discovery mempunyai 2 pin UART yaitu UART 4 dan UART 5, dan memiliki 4 pin USART yang bisa diatur untuk menjadi pin UART. Untuk konfigurasi pin secara keseluruhan dapat dilihat pada Gambar 2.14



Gambar 2.14 Pinout STM32F4 Discovery[9]

## 2.5 Purwarupa Mesin Cetak Huruf Braille ITS

Purwarupa Mesin Cetak Huruf Braille ITS dirancang dan dikembangkan oleh Tim Fakultas Teknologi Elektro ITS, dengan tim inti beranggotakan Dr. Tri Arief Sardjono, S.T., M.T., Ir. Tasripan, M.T., dan Dr. Ir. Hendra Kusuma, M.Eng.Sc. yang bekerja sama dengan Direktorat Pembinaan PK-LK Pendidikan Dasar dan Kementerian Pendidikan dan Kebudayaan. Purwarupa ini telah dikembangkan sejak tahun 2012 ini merupakan pengembangan dari mesin cetak dari Norwegia. Purwarupa Mesin Cetak Huruf Braille ITS ini memiliki fitur antara lain adalah,

- Daya rendah (hemat energi)
- Kemudahan dalam Pengoperasian dan Pemeliharaan
- Dapat dioperasikan oleh operator TUNANETRA
- Mampu mencetak karakter standard Braille 6 titik/dot sebanyak 1200 halaman/jam atau 400 karakter/detik
- Dapat mencetak langsung pada 2 sisi kertas (*double-sided*)
- Mempunyai kemampuan mencetak maksimum 42 karakter/baris
- Mempunyai komponen lokal hampir 85%



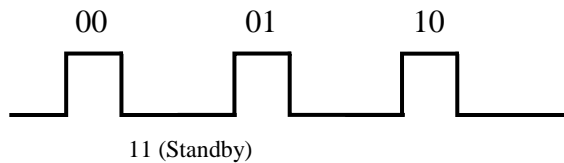
Gambar 2.15 Purwarupa Mesin Cetak Braille ITS

Purwarupa Mesin Cetak Huruf Braille ITS, memiliki antarmuka berupa tombol dan buzzer serta LCD 12x2 yang diintegrasikan dengan atmega. Untuk mengintegrasikan dengan perangkat antarmuka pada tugas akhir ini, purwarupa telah di modifikasi untuk dapat menerima 2 bit dari alat tugas akhir ini. Konfigurasi dari 2 bit ini dapat dilihat pada tabel dibawah.

Perintah Suara	Bit yang dikirimkan	Perintah pada purwarupa
-	11	Standby
“Test Print”	11	Standby
“Satu”	00	Test Print 1
“Dua”	01	Test Print 2
“Tiga”	10	Test Print 3

Tabel 2.2 Konfigurasi Perintah Purwarupa

Bit yang selalu masuk untuk membuat purwarupa ini berada pada keadaan standby adalah bit 11. Jika ada perintah yang mengirimkan bit “00”, “01”, “10” harus langsung dikembalikan ke bit 11 kembali untuk posisi standby sehingga perintah “Satu”, “Dua”, dan “Tiga” merupakan perintah berupa *trigger* saja.



Gambar 2.16 Proses Komunikasi pada Purwarupa

-----Halaman ini sengaja dikosongkan-----



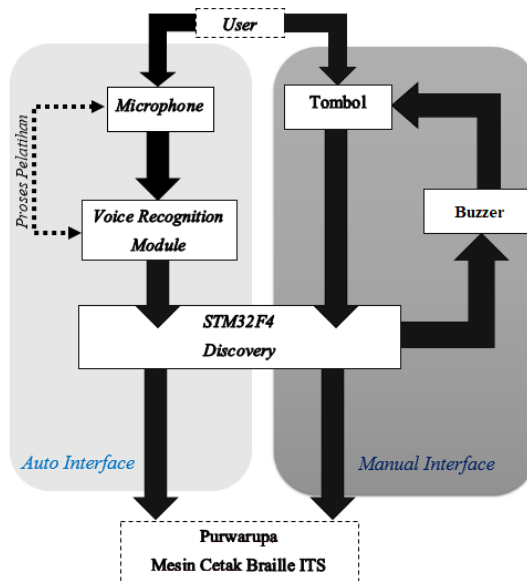
### BAB III

## PERANCANGAN PERANGKAT ANTARMUKA

Pada bab ini akan dibahas mengenai perancangan dan pembuatan sistem perangkat antarmuka dengan pengenalan suara untuk mengontrol purwarupa Mesin Cetak Huruf Braille ITS seperti diagram blok sistem, proses pelatihan modul, proses pengambilan data hex dari komunikasi serial, dan perancangan tombol dan diagram alur dari keseluruhan program pada papan tunggal STM32F4 *Discovery*.

### 3.1 Diagram Blok Sistem Antarmuka

Dalam tugas akhir ini, sistem yang dibuat adalah antarmuka untuk purwarupa Mesin Cetak Huruf Braille ITS menggunakan dua fitur yaitu pengenalan suara dan juga tombol dengan keluaran suara dari *buzzer* yang diproses menggunakan papan tunggal STM32F4 *Discovery*. Pada Gambar 3.1 dibawah dapat dilihat diagram blok dari keseluruhan sistem dari antarmuka ini.



Gambar 3.1 Diagram Blok Perangkat Sistem Antarmuka

Dapat dilihat dari diagram blok bahwa sistem antarmuka untuk mengendalikan purwarupa Mesin Cetak Huruf Braille ITS ini terbagi menjadi dua sistem yaitu sistem *Auto Interface* dan *Manual Interface* yang mana pada sistem *Auto* menggunakan kontrol dengan perintah pengelaran suara otomatis sedangkan pada sistem *interface* yang *Manual* menggunakan tombol yang memberi keluaran berupa suara sebelum digunakan untuk mengontrol purwarupa Mesin Cetak Huruf Braille ITS. Dari kedua sistem antarmuka ini jika digabungkan akan sangat membantu para penyandang tunanetra untuk mengoperasikan Mesin Cetak Huruf Braille ITS. Setelah ini akan dibahas secara detail mengenai proses perancangan kedua sistem tersebut

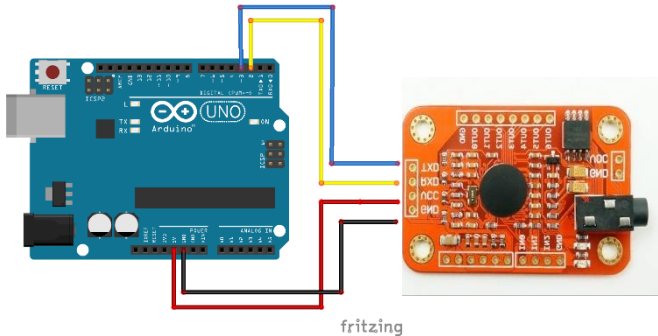
### **3.2 Perancangan Antarmuka Otomatis**

Pada perancangan sistem antarmuka otomatis, digunakan komponen berupa *Elechouse Voice Recognition Module V3* sebagai modul pengenalan suara nya, sebuah *electret condenser microphone* sebagai sensor suara, dan *FTDI Serial to USB* yang digunakan untuk memantau dan mengambil serta memindahkan data secara serial dari modul pengenalan suara menuju papan tunggal STM32F4 *Discovery* untuk selanjutnya diproses agar dapat menghasilkan keluaran berupa data 2 bit yang akan dikirim ke *board* pada purwarupa Mesin Cetak Huruf Braille ITS.

#### **3.2.1 Proses Pelatihan *Voice Recognition 3 Module***

Ini merupakan proses pertama yang dilakukan dalam membuat sistem antarmuka otomatis. Pada modul pengenalan suara bernama *Elechouse Voice Recognition 3 Module* harus dilakukan proses pelatihan awal agar modul dapat mengenali suara yang nantinya akan dijadikan *command* dalam mikrokontroller. Proses pelatihan modul ini dapat dilakukan dengan dua cara yaitu menggunakan *library* dan *sample program* dari Arduino atau bisa juga dengan cara komunikasi serial menggunakan serial terminal dengan bantuan *FTDI Serial to USB*.

Pada proses pelatihan digunakan *library* dan *sample program* dari Arduino, untuk itu modul harus dikoneksikan dengan Arduino terlebih dahulu seperti pada Gambar 3.2 yang ada dibawah ini.



fritzing

Gambar 3.2 Koneksi Arduino dengan Voice Recognition Module V3

```

vr_sample_train | Arduino 1.8.4
File Edit Sketch Tools Help

vr_sample_train
#include <SoftwareSerial.h>
#include "VoiceRecognitionV3.h"

/**
 * Connection
 * Arduino    VoiceRecognitionModule
 * 2  -----> TX
 * 3  -----> RX
 */
VR myVR(2,3); // 2:RX 3:TX, you can choose your favourite pins.

/***** declare print functions */
void printSeparator();
void printSignature(uint8_t *buf, int len);
void printVR(uint8_t *buf);
void printLoad(uint8_t *buf, uint8_t len);
void printTrain(uint8_t *buf, uint8_t len);
void printCheckRecognizer(uint8_t *buf);
void printUserGroup(uint8_t *buf, int len);

Done uploading.

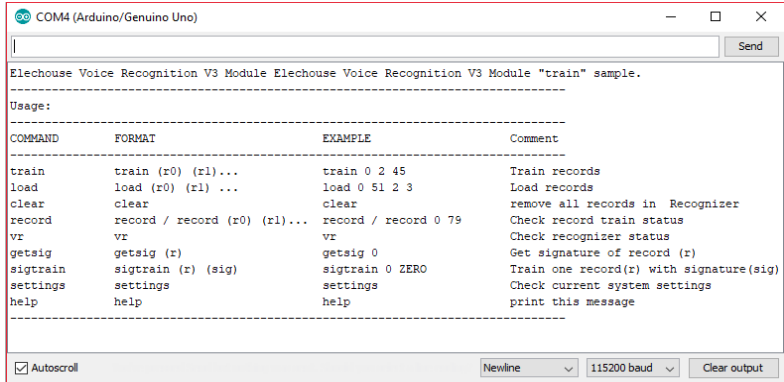
Sketch uses 11884 bytes (36%) of program storage space. Maximum is 32256 bytes.
Global variables use 901 bytes (43%) of dynamic memory, leaving 1147 bytes for local variables.

1 Arduino/Genuino Uno on COM4

```

Gambar 3.3 vr sample train.ino Untuk Melatih Modul

Setelah dikoneksikan seperti pada Gambar 3.2 lalu buka program bernama `vr_sample_train.ino` dengan menggunakan Arduino IDE. Setelah program tersebut *dicompile* kedalam Arduino lalu buka serial monitor pada Arduino untuk melakukan komunikasi serial terhadap modul. Pada serial monitor akan langsung otomatis memberikan perintah “*help*” dan akan dapat dilihat *command* apa saja yang dapat di gunakan dalam melatih modul *Voice Recognition V3* yang dapat dilihat pada Gambar 3.3.



Gambar 3.4 Daftar *command* Untuk Melatih Modul

Pada sistem *Auto Interface* ini digunakan 4 perintah suara yaitu kata-kata “Test Print”, “Satu”, “Dua”, “Tiga” dengan menggunakan admin suara pria dan wanita. Karena 4 perintah suara dengan dua kategori yaitu pria dan wanita bertotalkan 8 perintah suara yang harus dilatih sedangkan modul *Voice Recognition V3* hanya dapat menjalankan 7 perintah dalam waktu yang sama maka untuk masing-masing kategori jenis kelamin nantinya akan diuji satu per satu. Untuk pria diunggah dengan index 0 1 2 3 dan untuk wanita dengan index 10 11 12 13. Daftar kode pelatihan yang diberikan pada modul dapat dilihat pada Tabel 3.1 berikut :

Kategori	Index	Kata yang dilatih
Admin Pria	Sigtrain 0	Test Print
	Sigtrain 1	Satu
	Sigtrain 2	Dua
	Sigtrain 3	Tiga
Admin Wanita	Sigtrain 10	Test Print
	Sigtrain 11	Satu
	Sigtrain 12	Dua
	Sigtrain 13	Tiga

Tabel 3.1 Daftar Kata yang Dilatih

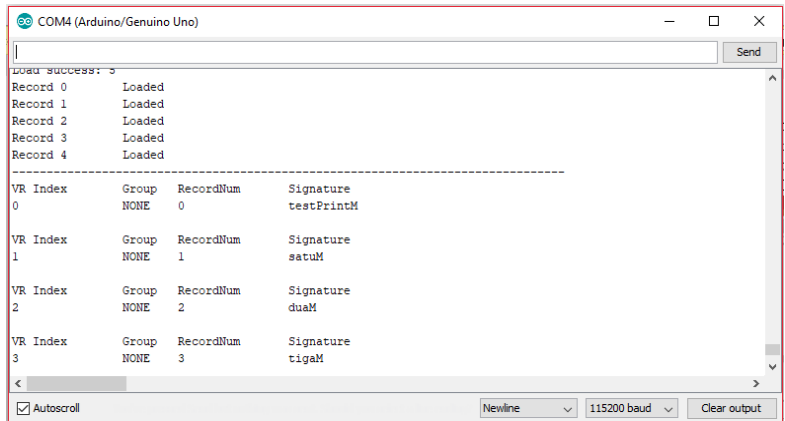
Proses pelatihan dimulai dengan memasukkan *command* sigtrain [nomor index] [signature]. Setelah modul membalas *Speak Now* pada serial monitor, ucapkan kata-kata yang akan dilatih pada index tersebut dan ucapkan lagi saat modul membalas *Speak Again*. Jika modul merasa kata yang diucapkan tidak cocok dengan kata pertama maka akan muncul balasan *Cann't Matched* dan dilakukan proses awal lagi sampai muncul balasan *Success*. Pada proses pelatihan untuk memasukkan *voice command* pada index tertentu ini hanya diberikan dua kali latihan pada saat perintah *Speak Now* dan *Speak Again*. Proses ini dapat dilihat pada Gambar 3.4.

Setelah proses pelatihan kata dilakukan maka proses selanjutnya adalah mengunggah index-index kata tadi kedalam modul. Untuk index yang sudah terlatih (*Trained*) tetapi tidak diunggah kedalam modul akan tetap tersimpan didalam modul tetapi tidak bisa diaktifkan karena keterbatasan modul. Setelah index-index yang diinginkan diunggah kedalam *recognizer*, baru pengenalan suara dapat dicoba. Jika suara terekognisi maka akan muncul index yang dikenali pada serial monitor seperti pada Gambar 3.5.

```
sigtrain 0 testPrintM
-----
Record: 0      Speak now
Record: 0      Speak again
Record: 0      Cann't matched
Record: 0      Speak now
Record: 0      Speak again
Record: 0      Success
Success: 1
Record 0       Trained
SIG: testPrintM
-----
sigtrain 1 satuM
-----
Record: 1      Speak now
Record: 1      Speak again
Record: 1      Success
Success: 1
Record 1       Trained
SIG: satuM
-----
sigtrain 2 duaM
-----
Record: 2      Speak now
Record: 2      Speak again
Record: 2      Success
Success: 1
Record 2       Trained
SIG: duaM
-----
sigtrain 3 tigaM
-----
Record: 3      Speak now
Record: 3      Speak again
Record: 3      Success
Success: 1
Record 3       Trained
SIG: tigaM
-----
```

☒ Autoscroll

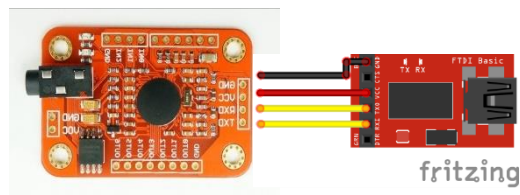
Gambar 3.5 Proses Pelatihan Kata



Gambar 3.6 Hasil Pengenalan Suara pada Serial Monitor

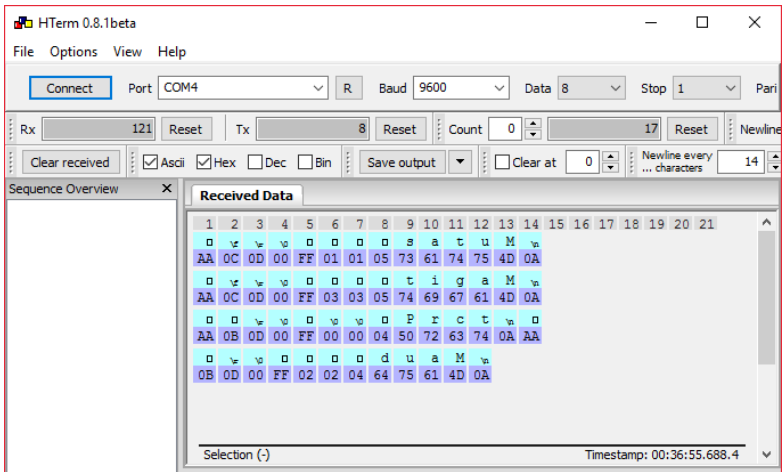
### 3.2.2 Proses Pengambilan Data Serial

Setelah modul dilatih dan index diunggah langkah selanjutnya yang dilakukan adalah mengetahui data serial yang dikirim oleh modul *Voice Recognition Module V3* saat modul tersebut merekognisi suatu suara yang cocok dengan index, karena modul ini terintegrasi dengan Arduino sedangkan dalam tugas akhir ini mikrokontroller yang digunakan adalah STM32F4 *Discovery*. Maka proses pengambilan data serial ini menggunakan *FTDI Serial to USB* dan dengan bantuan aplikasi serial monitor bernama *hterm*. Dapat dilihat pada Gambar 3.7 koneksi dari modul ke FTDI Serial to USB.



Gambar 3.7 Koneksi USB FTDI dengan *Voice Recognition Module V3*

Setiap kali *Voice Recognition Module V3* diputus dari sumber tegangan, modul akan secara otomatis menghapus semua index pada recognizer, maka dari itu setiap dinyalakan, modul harus diberikan perintah load terlebih dahulu. Pada modul dapat terlihat LED sebagai indikator, saat LED menyala menandakan pada modul tidak ada index yang masuk pada *recognizer* sehingga modul tidak berada pada posisi *stand by* untuk mengenali suara. Untuk membuat modul berada pada posisi *stand by* dan siap untuk merekognisi suara, masing-masing index yang dibutuhkan harus diunggah terlebih dahulu. Untuk menggunggah, modul harus menerima data serial hex dengan format | AA| 3+n | 30 | R0 | ... | Rn | 0A | dengan N merupakan index dari perintah yang ingin diunggah. Pengiriman data hex ini dilakukan oleh *FTDI Serial to USB* pada *hterm*. LED pada modul yang tadinya menyala akan menjadi *blink* ini menandakan modul sudah diunggah oleh index-index perintah dan *recognizer* pada modul siap merekognisi. Setelah melakukan rekognisi suara dapat dilihat bahwa hasil rekognisi suara modul adalah pengiriman data hexadecimal setiap kali salah satu index terekognisi. Dapat dilihat dari hasil pada tabel 3.2 dan gambar 3.5



Gambar 3.8 Data hex berdasarkan pengenalan pada *Hterm*



Kata yang dikenali	Data yang dikirim modul (Hex)
“Test Print”	AA 11 0D FF 00 00 0A 74 65 73 74 50 72 69 6E 74 4D 0A
“Satu”	AA 0C 0D 00 FF 01 01 05 73 61 74 75 4D 0A
“Dua”	AA 0B 0D 00 FF 02 00 04 64 75 61 4D 0A
“Tiga”	AA 0C 0D 00 FF 03 01 05 74 69 67 61 4D 0A

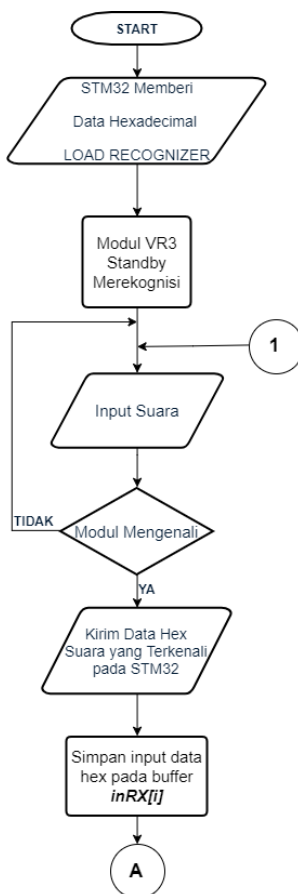
Tabel 3.2 Data Hex yang Dikirim Modul Berdasarkan Pengenalan suara Admin Pria

### 3.2.3 Flowchart Program Antarmuka Otomatis

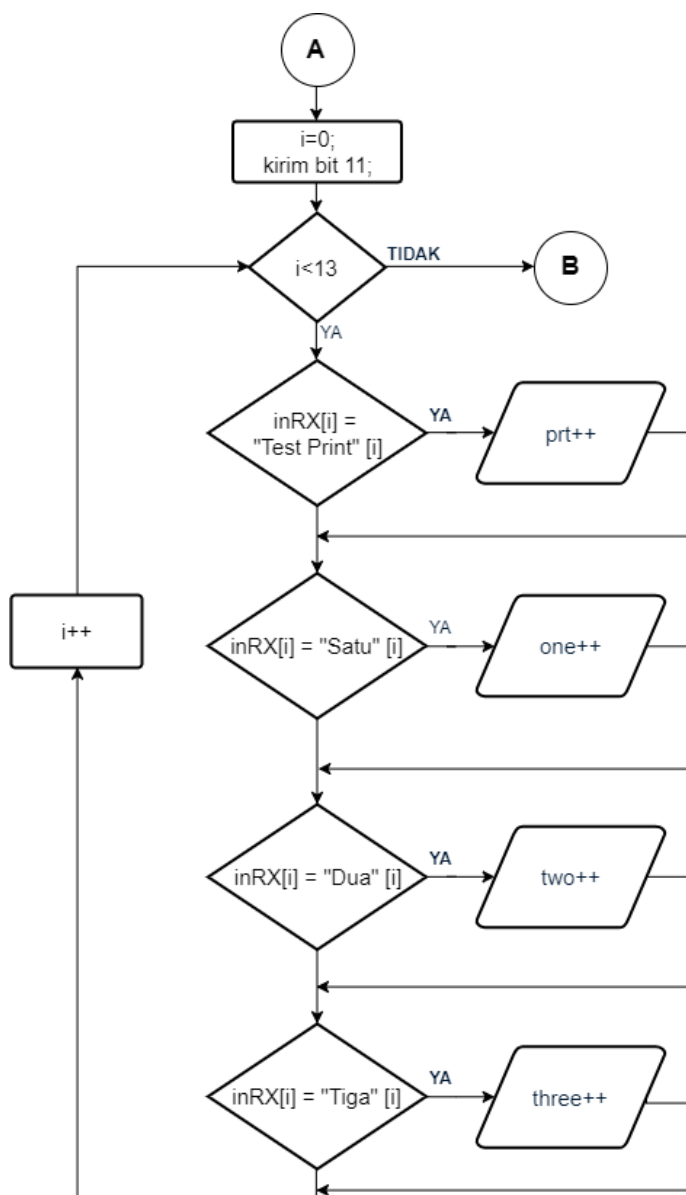
Sistem *Auto Interface* dibuat dengan memeriksa output yang diberikan oleh modul setelah modul mendapatkan input suara. Dengan proses pengambilan data serial pada 3.2.2 dapat diketahui masing-masing data yang dikirimkan oleh *Voice Recognition Module V3* pada setiap perintah yang direkognisi oleh modul. Dengan ini dapat dibuat database berupa buffer atau array untuk setiap data hex yang dikirim modul. Program diawali dengan inisialisasi UART, Clock, GPIO serta variabel variabel dan fungsi yang akan digunakan pada STM32F4-Discovery. Lalu dibuat sebuah fungsi yang dinamakan void recog() yang berisikan pengiriman data hex oleh pin tx STM32F4 kepada *Voice Recognition Module V3*. Data hex yang dikirimkan merupakan data *load index to recognizer* agar modul berada pada posisi *standby* untuk merekognisi suara lalu ada database berupa buffer untuk setiap data hex yang mungkin dikirim oleh modul saat merekognisi suara. Dan terdapat juga 4 variabel bebas masing-masing untuk setiap index suara. Saat input suara masuk dan terekognisi oleh modul, modul akan mengirimkan data hex sesuai dengan hasil rekognisinya, data inilah yang akan dicocokkan dengan database yang sudah ada. Masing-masing index database memeriksa input data yang masuk dari rx STM32F4. Proses awal ini dapat dilihat pada Gambar 3.9 (a).

Sistem pencocokan (*matching*) dan pengelompokan yang dibuat disini menggunakan cara membandingkan input data hexadecimal dengan database per bit nya. Pada saat modul mengirim input data, sistem melakukan pencocokan data input dengan ke-empat database. Secara sederhana, keempat-empatnya memiliki penghitung nya sendiri, setiap

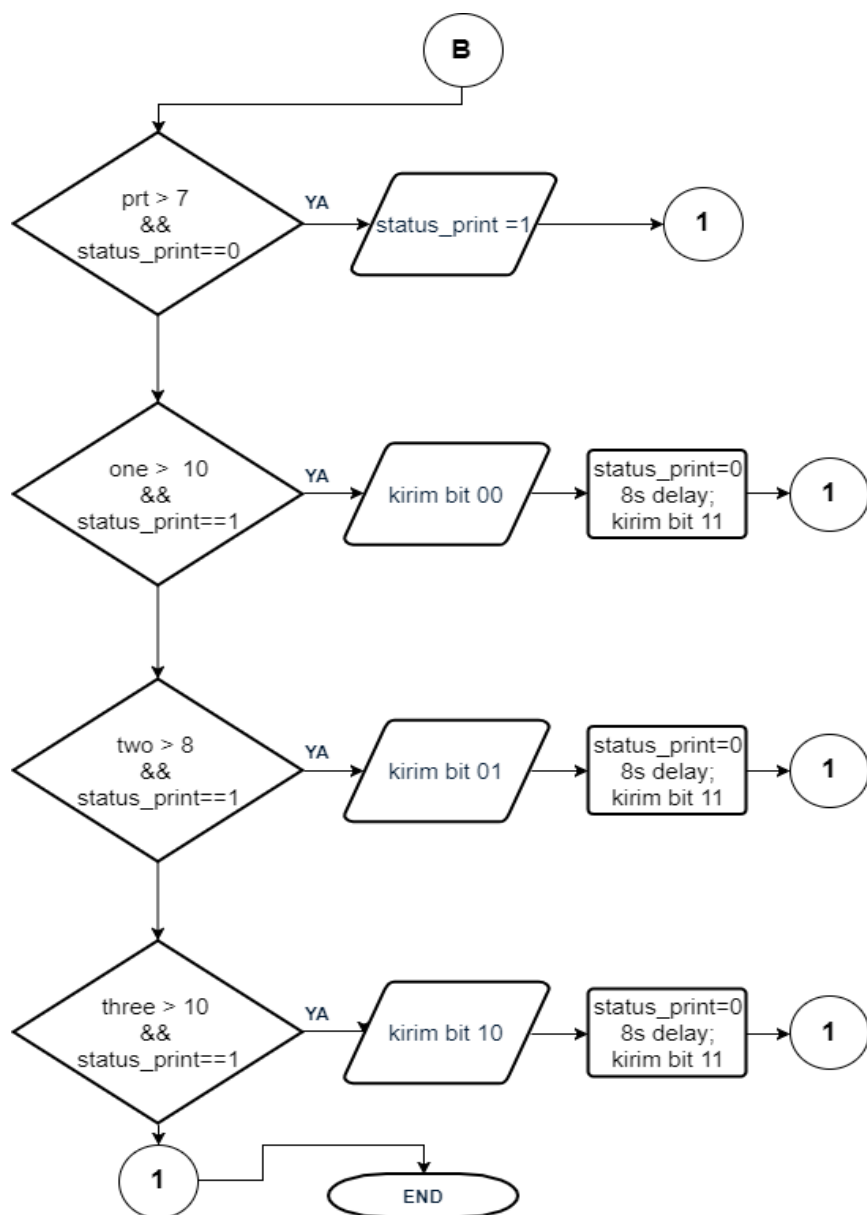
kali ada bit yang cocok dengan databasenya, penghitung nya akan bertambah satu. Penghitung ini merupakan variabel bebas bertipe integer. Lalu setelah semua penghitung dari masing-masing index menghitung bit yang sama dengan data input barulah masuk ke proses identifikasi, keempat penghitung dibandingkan, penghitung yang memiliki jumlah paling banyak berarti yang paling mirip dengan data input sehingga index dapat teridentifikasi. Dapat dilihat pada Gambar 3.9 (b)



Gambar 3.9 Flowchart Auto Interface (a)



Gambar 3.9 Flowchart Auto Interface (b)



Gambar 3.9 Flowchart Auto Interface (c)

Output yang diberikan oleh papan tunggal STM32F4-*Discovery* nantinya akan dikoneksikan menjadi input pada papan tunggal yang ada pada purwarupa Mesin Cetak Huruf Braille ITS. Komunikasi antara perangkat antarmuka dengan purwarupa Mesin Cetak menggunakan 2 bit data yang sudah dikonfigurasi pada purwarupa itu sendiri. Konfigurasinya dapat dilihat pada Tabel 3.3 dibawah ini:

Perintah Suara	Bit yang dikirimkan	Perintah pada purwarupa
-	11	Standby
“Test Print”	11	Standby
“Satu”	00	Test Print 1
“Dua”	01	Test Print 2
“Tiga”	10	Test Print 3

Tabel 3.3 Konfigurasi perintah dari STM32F4 ke purwarupa

Pengaturan 2 bit ini sudah disesuaikan dengan *board* pada purwarupa Mesin Cetak Huruf Braille ITS. Sebelum diberikan perintah suara apapun, STM32F4 memberikan bit 11 untuk memerintahkan purwarupa untuk standby, setelah diberikan perintah suara ”Test Print” bit yang dikirimkan juga tidak mengalami perubahan karena tetap pada posisi *standby*, perintah “Test Print” merupakan perintah pembuka untuk dapat dilanjutkan kepada perintah angka “Satu”, “Dua”, “Tiga”, hal ini dilakukan sebagai pengaman untuk mengatasi error pada modul pengenalan suara yang bisa saja tiba-tiba melakukan pengenalan suara yang tidak di inginkan. Jika kata “Satu” dikatakan setelah “Test Print” maka STM32F4 memberikan bit 00 yang hanya berupa *trigger* selama 8 detik kepada purwarupa lalu kemudian kembali ke bit 11 untuk posisi *standby* begitu juga hal yang sama dilakukan pada kata perintah “Dua” dan “Tiga”. Penjelasan ini dapat dilihat pada Gambar 3.9 (c)

### 3.3 Perancangan Antarmuka Manual

Pada perancangan manual digunakan 3 buah *push button*, *buzzer* dan papan tunggal STM32F4 *Discovery* sebagai mikrokontroler nya, *buzzer* digunakan untuk mengeluarkan suara saat tombol ditekan. Push button dikoneksikan dengan pin input pada STM32F4 dan pin output dikoneksikan pada *buzzer*. Karena terdapat 3 perintah yang harus dikomunikasikan terhadap purwarupa Mesin Cetak Huruf Braille ITS, maka dibutuhkan 3 buah *push button* dengan bunyi *buzzer* yang masing-masing berbeda untuk membedakan masing-masing tombol dengan suara.

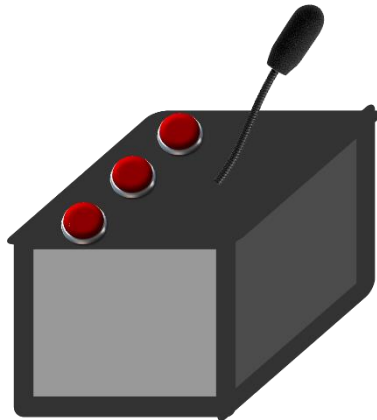
Perancangan ini menggunakan *counter* pada *push button* saat, *push button* ditekan kurang dari 3 detik maka *output* dari *push button* tersebut adalah *buzzer*, sedangkan saat *push button* ditekan lebih dari 3 detik maka *output* dari *push button* adalah komunikasi pada purwarupa. Bentuk komunikasi ketiga *push button* ini sama seperti pada perancangan antarmuka otomatis.

### 3.4 Perancangan *Hardware*

*Hardware* yang digunakan pada pembuatan perangkat antarmuka ini berupa :

1. Papan Tunggal STM32F4 *Discovery*
2. *Voice Recognition Module V3 by Elechouse with Microphone*
3. *Buzzer*

*Hardware* dirancang dengan bentuk balok yang didalamnya memuat papan tunggal STM32F4 *Discovery*, *Voice Recognition Module V3 by Elechouse*, dan 3 tombol Test Print.



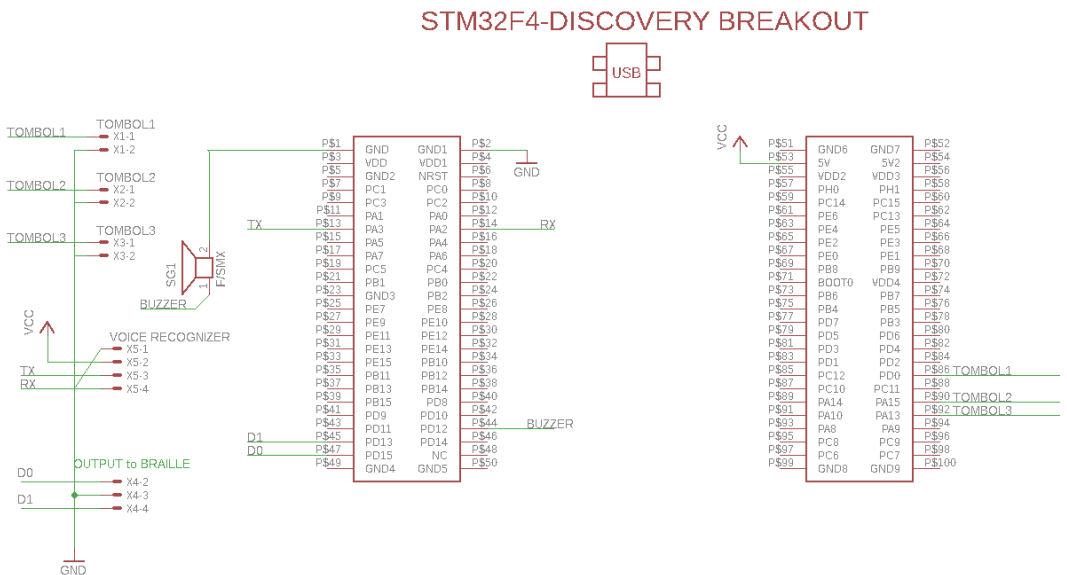
Gambar 3.10 Gambaran Desain *Hardware*

#### 3.4.2 Perancangan Perangkat Elektrik

Setelah *hardware* yang dibutuhkan sudah terkumpul, maka selanjutnya masing-masing komponen dikoneksikan, karena Papan Tunggal STM32F4 *Discovery* merupakan mikrokontroller pada

perancangan alat tugas akhir ini maka semua masukan dan keluaran dihubungkan dengan STM32F4 dapat dilihat pada Gambar 3.11

Pin yang digunakan adalah PA3 sebagai Tx dan PA2 sebagai Rx, PD13 dan PD15 adalah D0 dan D1 untuk dikonesikan pada Purwarupa Mesin Cetak Huruf Braille ITS, PD13 dan PD15 juga merupakan pin LED papan tunggal STM32F4 *Discovery*, sehingga dapat dipantau keluarannya melalui LED. PD12 dikoneksikan dengan *buzzer* dan juga merupakan pin LED pada papan tunggal STM32F4 *Discovery*. PD0, PA15, PA13 adalah pin *push button*



Gambar 3.11 Wiring Diagram Perangkat Antarmuka

-----*Halaman ini sengaja dikosongkan*-----



## **BAB IV**

### **PENGUJIAN DAN ANALISA PERANGKAT ANTARMUKA**

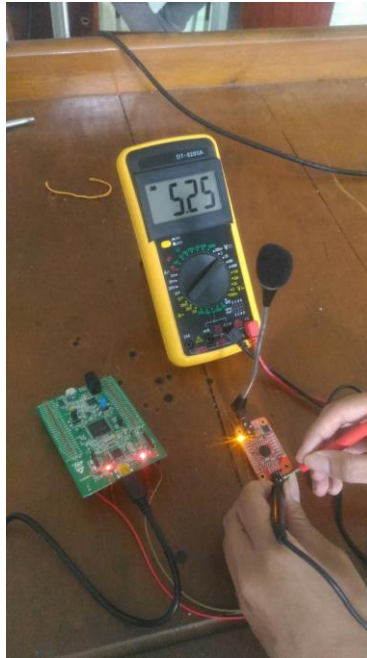
Pada bab ini akan dibahas mengenai pengujian dilakukan untuk mengetahui kinerja sistem secara keseluruhan. Pengujian dimulai dari pengujian penyuplai tegangan, pengambilan data sinyal suara dari modul yang sudah dilatih, pengujian komunikasi STM32F4 dengan purwarupa Mesin Cetak Huruf Braille ITS, dan pengujian keseluruhan sistem.

#### **4.1 Pengujian Supply Tegangan**

Rangkaian penyuplai tegangan berupa *charger handphone ASUS* dengan karakteristik *output 5.2 volt* dengan 1.5 Ampere. Bentuk fisik dari rangkaian penyuplai tegangan diperlihatkan pada Gambar 4.1. dan Gambar 4.2.



Gambar 4.1 Supply Tegangan STM32F4



Gambar 4.2 Supply Tegangan pada VR3 Module

Dapat dilihat bahwa tegangan yang tersupply pada STM32F4 yang kemudian di jumper ke modul *Voice Recognition Module V3* memberikan hasil 5.20 dan 5.25 Volt. Cocok dengan karakteristik dari *charger handphone ASUS*.

#### 4.2 Pengujian Pengenalan Suara pada Antarmuka Otomatis

Pada tahap pengujian pengenalan suara oleh *Voice Recognition V3 Module by Elechouse* dilakukan menggunakan Arduino untuk menguji keberhasilan rekognisi suara pada modul. Pengujian dilakukan dengan 3 kali percobaan yaitu, 8 orang pria diuji dengan database admin pria, 8 orang wanita diuji dengan referensi database admin wanita. Lalu 3 orang pria dan wanita diuji dengan referensi database nya masing-masing. Pria dan wanita yang menjadi penguji berumur 20-22 tahun

Dari ketiga cara uji coba tersebut (Pria – Admin, Wanita – Admin, Pria & Wanita – Sendiri) didapat sebuah hasil dimana rata-rata persentase tingkat keberhasilan dari sistem *Auto Interface* yang menggunakan modul

*Voice Recognition V3 Module by Elechouse* adalah seperti yang ditampilkan pada Tabel 4.1 berikut

Uji Coba	Rata-rata Persentase Keberhasilan
<b>Pria-Admin</b>	55,63%
<b>Wanita-Admin</b>	66,25%
<b>Referensi Orang Lain</b>	60,94%
<b>Admin-Admin</b>	77,92%

Tabel 4.1 Rata-Rata Persentase Keberhasilan

#### 4.2.1 Uji Coba Pria dengan Referensi Database Admin Pria

Uji coba ini dilakukan dengan database admin pria yang sudah tersimpan pada *Voice Recognition V3 Module*. Kemampuan rekognisi dari modul dites dengan 10 suara pria yang berbeda. Dari kesepuluh pria ini salah satu nya termasuk admin. Untuk pengujian dengan kata “Test Print” dilakukan sejumlah 10 kali untuk setiap penguji dan didapatkan hasil seperti pada Tabel 4.2

Nama Penguji	Jumlah berhasil
<b>Nico (Admin)</b>	10
<b>Anggarjuna</b>	1
<b>Andika</b>	5
<b>Gery</b>	0
<b>Hery</b>	0
<b>Richard</b>	4
<b>Rheza</b>	2
<b>Rizal</b>	0
<b>Persentase Keberhasilan</b>	27,5%

Tabel 4.2 Jumlah kata “Test Print” yang Berhasil Terekognisi dari 10 kali pengucapan

Untuk pengujian dengan kata “Satu” dilakukan sejumlah 10 kali untuk setiap penguji dan didapatkan hasil seperti pada Tabel 4.3

<b>Nama Penguji</b>	<b>Jumlah berhasil</b>
<b>Nico (Admin)</b>	10
<b>Anggarjuna</b>	4
<b>Andika</b>	1
<b>Gery</b>	0
<b>Hery</b>	5
<b>Richard</b>	10
<b>Rheza</b>	0
<b>Rizal</b>	0
<b>Persentase Keberhasilan</b>	<b>37,5%</b>

Tabel 4.3 Jumlah kata “Satu” yang Berhasil Terekognisi dari 10 kali pengucapan

Untuk pengujian dengan kata “Dua” dilakukan sejumlah 10 kali untuk setiap penguji dan didapatkan hasil seperti pada Tabel 4.4

<b>Nama Penguji</b>	<b>Jumlah berhasil</b>
<b>Nico (Admin)</b>	10
<b>Anggarjuna</b>	10
<b>Andika</b>	10
<b>Gery</b>	10
<b>Hery</b>	10
<b>Richard</b>	9
<b>Rheza</b>	1
<b>Rizal</b>	10
<b>Persentase Keberhasilan</b>	<b>87,5%</b>

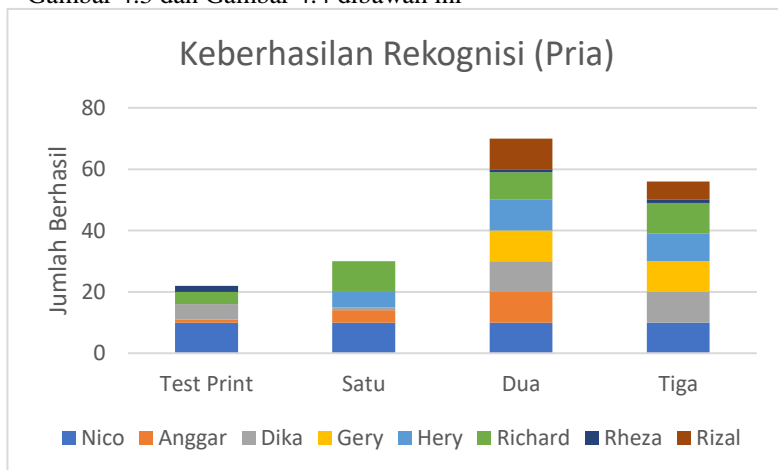
Tabel 4.4 Jumlah kata “Dua” yang Berhasil Terekognisi dari 10 kali pengucapan

Untuk pengujian dengan kata “Tiga” dilakukan sejumlah 10 kali untuk setiap penguji dan didapatkan hasil seperti pada Tabel 4.5

Nama Penguji	Jumlah berhasil
<b>Nico (Admin)</b>	10
<b>Anggarjuna</b>	0
<b>Andika</b>	10
<b>Gery</b>	10
<b>Hery</b>	9
<b>Richard</b>	10
<b>Rheza</b>	1
<b>Rizal</b>	6
<b>Persentase Keberhasilan</b>	<b>70%</b>

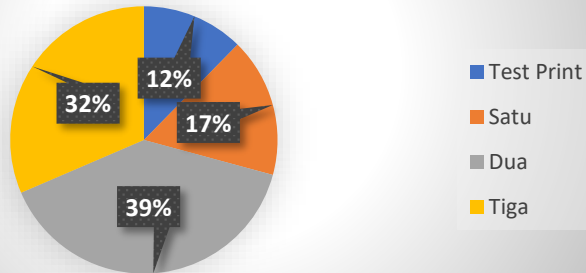
Tabel 4.5 Jumlah kata “Tiga” yang Berhasil Terekognisi dari 10 kali pengucapan

Dari data diatas dapat dirangkum menjadi diagram yang ditunjukkan oleh Gambar 4.3 dan Gambar 4.4 dibawah ini



Gambar 4.3 Diagram Batang Keberhasilan Rekognisi Perintah Suara pada Pria

## Persentase Keberhasilan Rekognisi (Pria)



Gambar 4.4 Diagram Lingkaran Persentase Keberhasilan Rekognisi Perintah Suara oleh *Voice Recognition Module V3* pada pria

Berdasarkan data diatas, kata-kata “Test Print” merupakan kata yang paling sulit untuk direkognisi. Sedangkan kata “Dua” merupakan kata paling mudah untuk direkognisi, hal ini dapat dikatakan sesuai dengan perancangan awal *Auto Interface* yang mana perintah “Test Print” digunakan untuk menjadi pengaman dari perintah suara lainnya, karena data persentase keberhasilan rekognisi perintah “Test Print” yang paling sedikit (12%). Lalu didapat kesimpulan bahwa rata-rata tingkat keberhasilan modul menggunakan suara pria dengan referensi admin (orang lain) adalah 55,63%

Kata Perintah Suara	Persentase Keberhasilan
Test Print	27,5%
Satu	37,5%
Dua	87,5%
Tiga	70%
<b>Rata-rata Persentase Keberhasilan</b>	<b>55,63%</b>

Tabel 4.6 Rata-Rata Persentase Keberhasilan Pria dengan Referensi Admin

#### 4.2.2 Uji Coba Wanita dengan Referensi Database Admin Wanita

Uji coba ini dilakukan dengan database admin wanita yang sudah tersimpan pada *Voice Recognition V3 Module*. Kemampuan rekognisi dari modul diuji dengan 10 suara pria yang berbeda. Dari kesepuluh wanita ini salah satu nya termasuk admin. Untuk pengujian dengan kata “Test Print” dilakukan sejumlah 10 kali untuk setiap penguji dan didapatkan hasil seperti pada Tabel 4.7

Nama Penguji	Jumlah berhasil
<b>Regita (Admin)</b>	10
<b>Sierly</b>	10
<b>Putri</b>	7
<b>Monic</b>	5
<b>Qumi</b>	2
<b>Della</b>	0
<b>Nadia</b>	10
<b>Cut</b>	1
<b>Persentase Keberhasilan</b>	56,25%

Tabel 4.7 Jumlah kata “Test Print” yang Berhasil Terekognisi dari 10 kali pengucapan

Untuk pengujian dengan kata “Satu” dilakukan sejumlah 10 kali untuk setiap penguji dan didapatkan hasil seperti pada Tabel 4.8

Nama Penguji	Jumlah berhasil
<b>Regita (Admin)</b>	10
<b>Sierly</b>	8
<b>Putri</b>	0
<b>Monic</b>	8
<b>Qumi</b>	6
<b>Della</b>	3
<b>Nadia</b>	10
<b>Cut</b>	0
<b>Persentase Keberhasilan</b>	56,25%

Tabel 4.8 Jumlah kata “Satu” yang Berhasil Terekognisi dari 10 kali pengucapan

Untuk pengujian dengan kata “Dua” dilakukan sejumlah 10 kali untuk setiap penguji dan didapatkan hasil seperti pada Tabel 4.9

Nama Penguji	Jumlah berhasil
<b>Regita (Admin)</b>	10
<b>Sierly</b>	10
<b>Putri</b>	10
<b>Monic</b>	8
<b>Qumi</b>	4
<b>Della</b>	9
<b>Nadia</b>	9
<b>Cut</b>	7
<b>Persentase Keberhasilan</b>	82,50%

Tabel 4.9 Jumlah kata “Dua” yang Berhasil Terekognisi dari 10 kali pengucapan

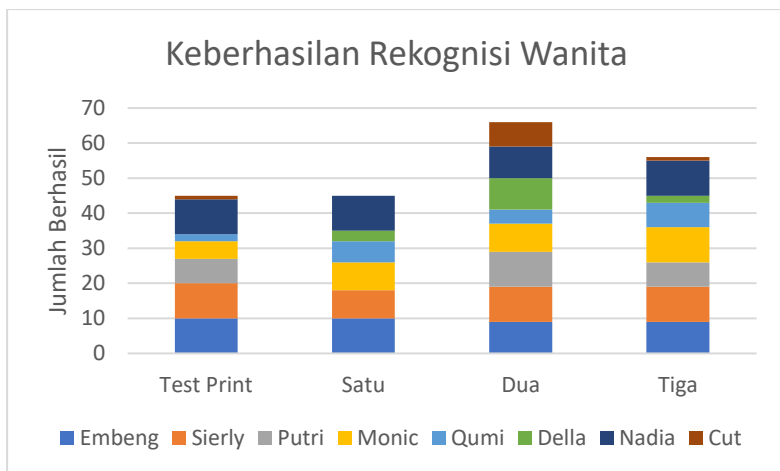
Untuk pengujian dengan kata “Tiga” dilakukan sejumlah 10 kali untuk setiap penguji dan didapatkan hasil seperti pada Tabel 4.10

Nama Penguji	Jumlah berhasil
<b>Regita (Admin)</b>	10
<b>Sierly</b>	10
<b>Putri</b>	7
<b>Monic</b>	10
<b>Qumi</b>	7
<b>Della</b>	2
<b>Nadia</b>	10
<b>Cut</b>	1
<b>Persentase Keberhasilan</b>	70%

Tabel 4.10 Jumlah kata “Tiga” yang Berhasil Terekognisi dari 10 kali pengucapan

Dari data diatas dapat dirangkum menjadi diagram yang ditunjukkan oleh Gambar 4.5 dan Gambar 4.6 dibawah ini





Gambar 4.5 Diagram Batang Keberhasilan Rekognisi Perintah Suara pada Wanita



Gambar 4.6 Diagram Lingkaran Persentase Keberhasilan Rekognisi Perintah Suara oleh *Voice Recognition Module V3* pada wanita

Berdasarkan data diatas, kata-kata “Test Print” masih merupakan kata yang paling sulit untuk direkognisi dan juga kata “Satu”. Sedangkan kata “Dua” merupakan kata paling mudah untuk direkognisi, sama seperti pada data keberhasilan rekognisi pada pria. Hal ini dapat dikatakan sesuai

dengan perancangan awal *Auto Interface* yang mana perintah “Test Print” digunakan untuk menjadi pengaman dari perintah suara lainnya, karena data persentase keberhasilan rekognisi perintah “Test Print” yang paling sedikit (56,25%) walaupun kata “Satu” juga memiliki presentasi yang sama (56,25%). Lalu didapat kesimpulan bahwa rata-rata tingkat keberhasilan modul menggunakan suara wanita dengan referensi admin (orang lain) adalah 66,25%

Kata Perintah Suara	Persentase Keberhasilan
<b>Test Print</b>	56,25%
<b>Satu</b>	56,25%
<b>Dua</b>	82,5%
<b>Tiga</b>	70%
<b>Rata-rata Persentase Keberhasilan</b>	66,25%

Tabel 4.11 Rata-Rata Persentase Keberhasilan Wanita dengan Referensi Admin

#### 4.2.3 Uji Coba dengan Referensi Sendiri

Uji coba ini dilakukan dengan sebelumnya melatih suara masing-masing penguji pada *Voice Recognition V3 Module*. Terdapat 6 penguji yang suaranya sudah dilatih kedalam modul tiga pria dan tiga wanita. Untuk pengujian dengan kata “Test Print” dilakukan sejumlah 10 kali untuk setiap penguji dan dibandingkan dengan jumlah berhasil yang sudah didapat sebelumnya menggunakan referensi admin didapatkan hasil seperti pada Tabel 4.12

Nama Penguji	Jumlah berhasil (Referensi Sendiri)	Jumlah Berhasil (Referensi Admin)
<b>Rheza</b>	8	2
<b>Rizal</b>	6	0
<b>Richard</b>	5	4
<b>Qumi</b>	7	2
<b>Della</b>	3	0
<b>Nadia</b>	10	10
<b>Persentase Keberhasilan</b>	65%	30%

Tabel 4.12 Jumlah kata “Test Print” yang Berhasil Terekognisi dari 10 kali pengucapan

Untuk pengujian dengan kata “Satu” dilakukan sejumlah 10 kali untuk setiap penguji dan didapatkan hasil seperti pada Tabel 4.13

<b>Nama Penguji</b>	<b>Jumlah berhasil (Referensi Sendiri)</b>	<b>Jumlah Berhasil (Referensi Admin)</b>
<b>Rheza</b>	10	0
<b>Rizal</b>	10	0
<b>Richard</b>	10	10
<b>Qumi</b>	8	6
<b>Della</b>	9	3
<b>Nadia</b>	10	10
<b>Persentase Keberhasilan</b>	95%	48,33%

Tabel 4.13 Jumlah kata “Satu” yang Berhasil Terekognisi dari 10 kali pengucapan

Untuk pengujian dengan kata “Dua” dilakukan sejumlah 10 kali untuk setiap penguji dan didapatkan hasil seperti pada Tabel 4.14

<b>Nama Penguji</b>	<b>Jumlah berhasil (Referensi Sendiri)</b>	<b>Jumlah berhasil (Referensi Admin)</b>
<b>Rheza</b>	10	1
<b>Rizal</b>	9	10
<b>Richard</b>	10	9
<b>Qumi</b>	8	4
<b>Della</b>	10	9
<b>Nadia</b>	9	9
<b>Persentase Keberhasilan</b>	93,33%	70%

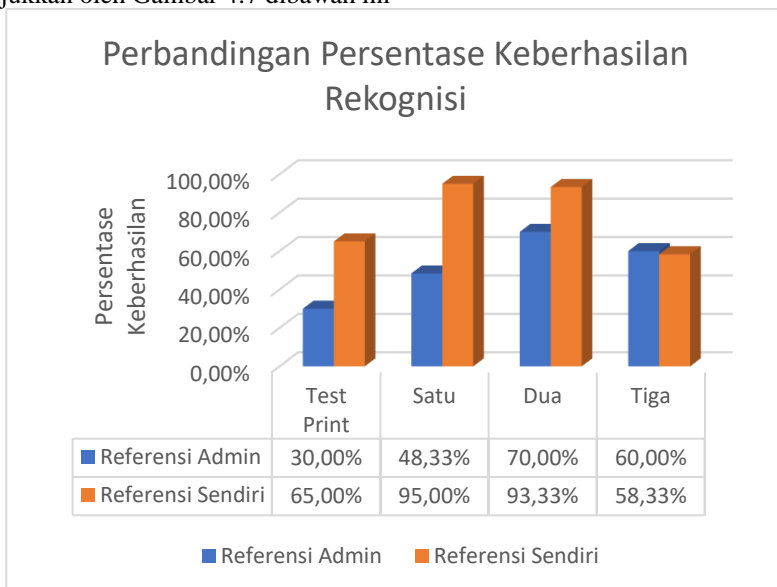
Tabel 4.14 Jumlah kata “Dua” yang Berhasil Terekognisi dari 10 kali pengucapan

Untuk pengujian dengan kata “Tiga” dilakukan sejumlah 10 kali untuk setiap penguji dan didapatkan hasil seperti pada Tabel 4.15

Nama Penguji	Jumlah berhasil (Referensi Sendiri)	Jumlah berhasil (Referensi Admin)
<b>Rheza</b>	3	1
<b>Rizal</b>	5	6
<b>Richard</b>	8	10
<b>Qumi</b>	7	7
<b>Della</b>	2	2
<b>Nadia</b>	10	10
<b>Persentase Keberhasilan</b>	58,33%	60%

Tabel 4.15 Jumlah kata “Tiga” yang Berhasil Terekognisi dari 10 kali pengucapan

Dari data diatas dapat dirangkum menjadi diagram yang ditunjukkan oleh Gambar 4.7 dibawah ini



Gambar 4.7 Diagram Batang Perbandingan Persentase Keberhasilan Rekognisi

Setelah melihat data diatas dapat disimpulkan bahwa hasil dari pengujian menggunakan referensi suara sendiri hasilnya akan jauh lebih bagus, walaupun terdapat sedikit penurunan pada kata “Tiga” dari 60% menjadi 58,33% tetapi dapat dilihat pada Tabel 4.16 hal ini dikarenakan ada dua orang yang mengalami penurunan pada kata “Tiga” setelah menggunakan referensi sendiri dan penurunan ini tidak terlalu signifikan sehingga masih dalam batas wajar. Lalu didapat kesimpulan bahwa rata-rata tingkat keberhasilan modul dengan menggunakan referensi suara sendiri adalah 77,92%

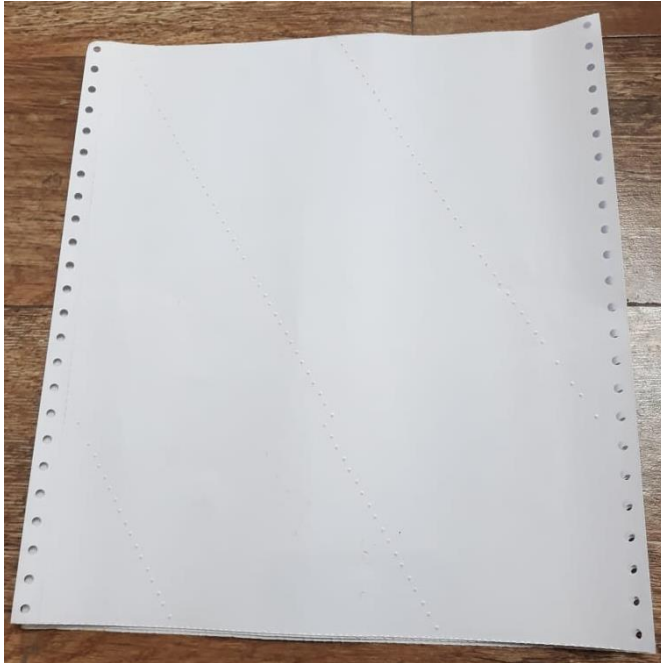
Kata Perintah Suara	Persentase Keberhasilan
<b>Test Print</b>	65%
<b>Satu</b>	95%
<b>Dua</b>	93,33%
<b>Tiga</b>	58,33%
<b>Rata-rata Persentase Keberhasilan</b>	77,92%

Tabel 4.16 Rata-Rata Persentase Keberhasilan Wanita dengan Referensi Admin

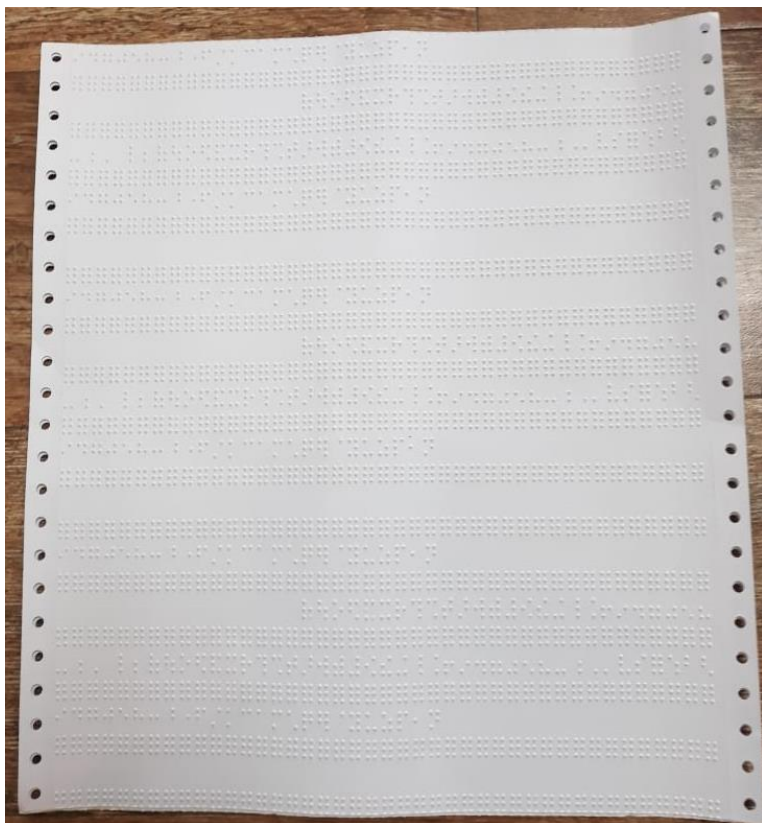
Pada pengujian menggunakan modul *Voice Recognition V3 Module by Elechouse* didapatkan bahwa persentase keberhasilan menggunakan referensi suara sendiri memiliki jumlah yang lebih besar dibanding pengujian menggunakan referensi suara orang lain. Namun dalam pengujian menggunakan referensi suara sendiri juga memiliki sedikit error, hal ini dikarenakan oleh beberapa faktor lain yaitu intonasi dan volume suara serta jarak penguji terhadap mikrofon. Hal ini terjadi juga dalam penelitian “*Testing of Other Languages Usage in Addition to the Default Languages for the Easy Voice Recognition Module*” oleh Rustam Asnawi dan Muhammad Said yang menggunakan modul pengenalan suara *Easy VR 3.0* bahwa modul mengenali suara melalui pengenalan perintah suara dan pengenalan dari intonasi suara. Orang yang tahu perintah suara dan tidak mengucapkan perintah suara dengan intonasi yang tepat seperti pada referensi, akan ditolak oleh sistem[10]. Dari hal ini dapat diketahui bahwa intonasi sangat berpengaruh pada beberapa modul pengenalan suara.

#### **4.3 Pengujian Komunikasi antara Perangkat Antarmuka dengan Purwarupa Mesin Cetak Huruf Braille ITS**

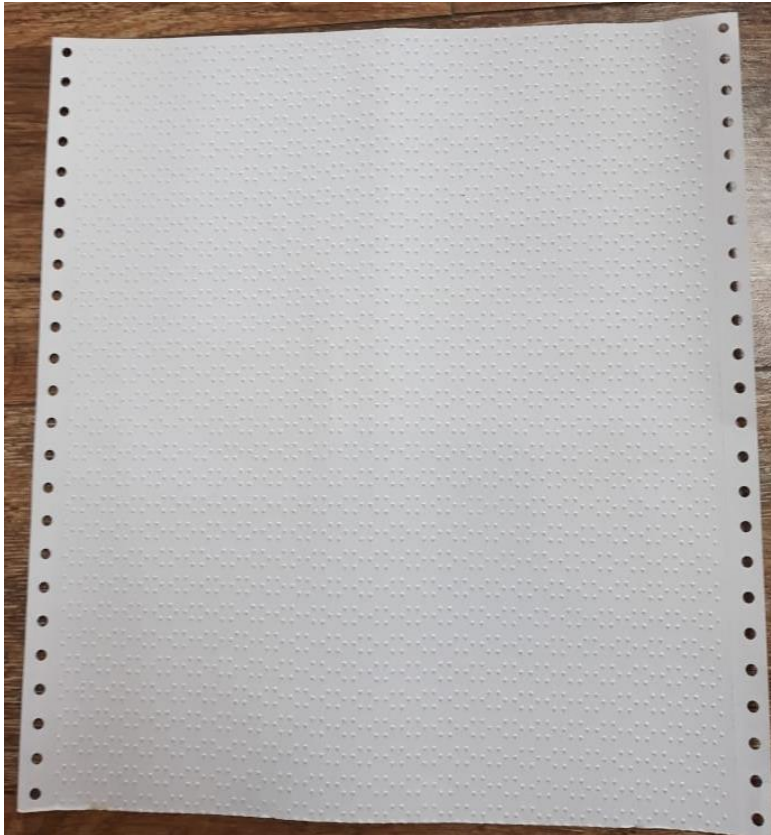
Pada purwarupa Mesin Cetak Huruf Braille ITS, terdapat 3 pola test print, pola ini dapat dilihat pada gambar 4.x, 4.x, 4.x



Gambar 4.8 Pola Test Print 1



Gambar 4.9 Pola Test Print 2



Gambar 4.10 Pola Test Print 3

Pengujian komunikasi pada purwarupa ini dilakukan oleh Pak Tasripan yang suaranya sudah terlebih dahulu dilatih kedalam sistem, hasil uji coba dapat dilihat pada tabel 4.x dibawah ini

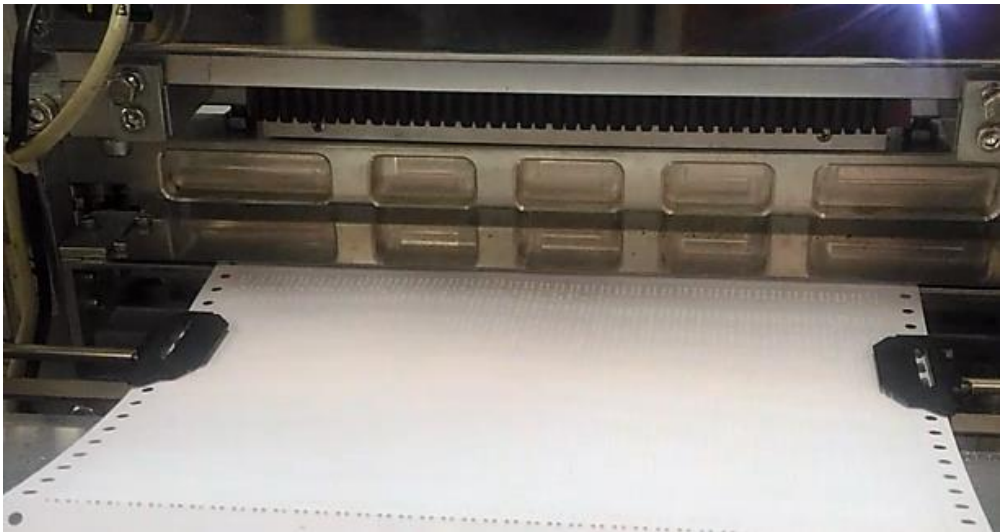
Perintah Suara	Hasil
Test Print – Satu	Gambar 4.11
Test Print – Dua	Gambar 4.12
Test Print – Tiga	Gambar 4.13

Tabel 4.17 Hasil Komunikasi pada Purwarupa

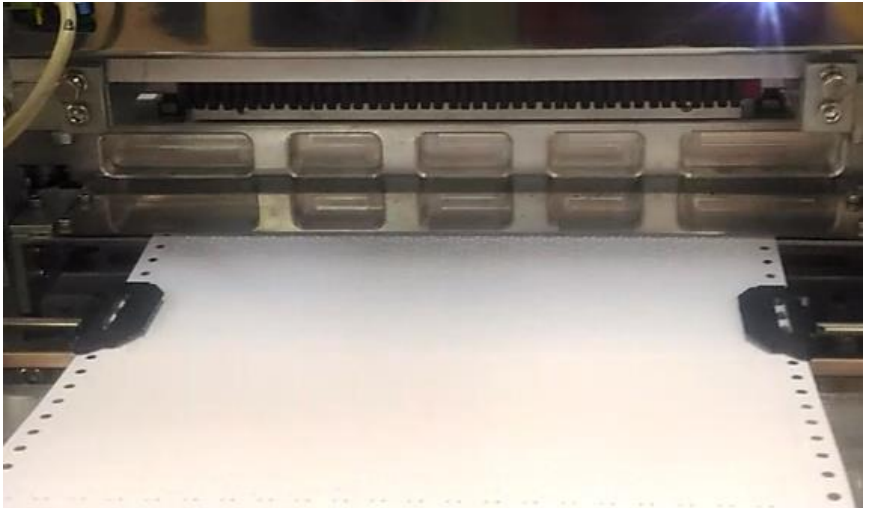




Gambar 4.11 Hasil Test Print 1



Gambar 4.12 Hasil Test Print 2



Gambar 4.13 Hasil Test Print 3

Pada hasil pengujian masing-masing perintah suara sudah dapat dilihat bahwa semua perintah suara sudah dapat berkomunikasi dengan baik dengan menjalankan perintah nya dengan benar pada purwarupa yang ditunjukkan pada Gambar 4.11, Gambar 4.12, Gambar 4.13.

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Sistem *interface hardware* yang dibuat dengan dua sistem *Auto* dan *Manual Interface* mampu berjalan dengan baik dengan keberhasilan 60,94% dengan referensi suara orang lain dan 77,92% referensi suara sendiri, namun ada beberapa hal yang mempengaruhi keberhasilan yaitu kondisi ruangan sekitar, semakin ramai kondisi ruangan akan semakin sulit modul mampu dalam mengenali suara. Intonasi dan volume suara, volume suara yang lebih tegas dan keras saat proses pelatihan modul akan lebih mudah dikenali nantinya saat diuji, intonasi juga sangat berpengaruh, dengan *user* yang sama tetapi mengubah nada intonasi juga sedikit lebih susah dikenali oleh modul, begitupula *user* yang berbeda tetapi berusaha memiripkan nada atau intonasi dari suara referensinya akan sedikit lebih mudah untuk dikenali. Dan berdasarkan data tingkat keberhasilan dapat dilihat referensi dengan suara sendiri lebih akurat daripada dengan referensi suara orang lain.

#### **5.2 Saran**

Beberapa saran yang dapat dilakukan untuk penelitian yang lebih lanjut dari tugas akhir ini adalah:

1. Menggunakan modul pengenalan suara yang lebih bisa dikontrol dan di deteksi data suaranya.
2. Menggunakan filter analog maupun digital sebelum memasuki proses latihan pada modul.
3. Menggunakan metode pengenalan suara yang lebih bagus agar sistem dapat mengadaptasi setiap input baru sebagai proses latihan.
4. Mempelajari bentuk sinyal suara pada artikulasi kata yang diucapkan.

-----*Halaman ini sengaja dikosongkan*-----

## DAFTAR PUSTAKA

- [1] “ITS Siap Pasarkan Mesin Cetak Braille,” *ITS News*, 26-Nov-2017.
- [2] Z. Zhang, “Mechanics of human voice production and control,” *J. Acoust. Soc. Am.*, vol. 140, no. 4, hlm. 2614–2635, Okt 2016.
- [3] K. Honda, “Physiological Processes of Speech Production,” dalam *Springer Handbook of Speech Processing*, J. Benesty, M. M. Sondhi, dan Y. A. Huang, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, hlm. 7–26.
- [4] P. Rubin dan E. Vatikiotis-Bateson, “Measuring and Modeling Speech Production,” dalam *Animal Acoustic Communication: Sound Analysis and Research Methods*, S. L. Hopp, M. J. Owren, dan C. S. Evans, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, hlm. 251–290.
- [5] S. Narang dan D. Gupta, “Speech Feature Extraction Techniques: A Review,” hlm. 8, 2015.
- [6] T. Khdour, Ph. Muaidi, A. Al-Ahmad, S. Algrainy, dan M. Alkoffash, *Arabic Audio News Retrieval System Using Dependent Speaker Mode, Mel Frequency Cepstral Coefficient and Dynamic Time Warping Techniques*, vol. 7. 2014.
- [7] “Speak Recognition, Voice Recognition Module V3 [AD\_VR3] - \$28.00 : Elechouse, Arduino Play House.” [Daring]. Tersedia pada: [http://www.elechouse.com/elechouse/index.php?main\\_page=product\\_info&cPath=168\\_170&products\\_id=2254&zenid=j2e8jiv0992ji177gmj3fvbhc2](http://www.elechouse.com/elechouse/index.php?main_page=product_info&cPath=168_170&products_id=2254&zenid=j2e8jiv0992ji177gmj3fvbhc2). [Diakses: 19-Jun-2018].
- [8] “STM32F4DISCOVERY - Discovery kit with STM32F407VG MCU \* New order code STM32F407G-DISC1 (replaces STM32F4DISCOVERY) - STMicroelectronics.” [Daring]. Tersedia pada: [http://www.st.com/content/st\\_com/en/products/evaluation-tools/product-evaluation-tools/mcu-eval-tools/stm32-mcu-eval-tools/stm32-mcu-discovery-kits/stm32f4discovery.html](http://www.st.com/content/st_com/en/products/evaluation-tools/product-evaluation-tools/mcu-eval-tools/stm32-mcu-eval-tools/stm32-mcu-discovery-kits/stm32f4discovery.html). [Diakses: 19-Jun-2018].
- [9] “stm32f4 discovery pinout,” *Virtual Pinball Universe*. [Daring]. Tersedia pada: <http://vpuniverse.com/forums/gallery/image/572-stm32f4-discovery-pinout/>. [Diakses: 24-Jun-2018].
- [10] R. Asnawi dan M. Said, “Testing of other languages usage in addition to the default languages for the easy voice recognition module,” hlm. 4.

-----*Halaman ini sengaja dikosongkan*-----



## LAMPIRAN

### A. Tampak Atas Perangkat Antarmuka



B. Tampak Depan Purwarupa Mesin Cetak Huruf Braille  
ITS



## C. Program pada Keil $\mu$ Vision 5 IDE

```
1.  /** *****
*****  * @file           : main.c  * @brief           : Main program body
*****
****  * This notice applies to any and all portions of this file  * that
are not between comment pairs USER CODE BEGIN and  * USER CODE END. Other p
ortions of this file, whether  * inserted by the user or by software devel
opment tools  * are owned by their respective copyright owners.  * * COPYR
IGHT(c) 2018 STMicroelectronics  * * Redistribution and use in source and
binary forms, with or without modification,  * are permitted provided that
the following conditions are met:  * 1. Redistributions of source code mu
st retain the above copyright notice,  * this list of conditions and t
he following disclaimer.  * 2. Redistributions in binary form must reprod
uce the above copyright notice,  * this list of conditions and the fol
lowing disclaimer in the documentation  * and/or other materials provi
ded with the distribution.  * 3. Neither the name of STMicroelectronics n
or the names of its contributors  * may be used to endorse or promote
products derived from this software  * without specific prior written
permission.  * * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CO
NTRIBUTORS "AS IS"  * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, THE  * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS F
OR A PARTICULAR PURPOSE ARE  * DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
HOLDER OR CONTRIBUTORS BE LIABLE  * FOR ANY DIRECT, INDIRECT, INCIDENTAL, S
PECIAL, EXEMPLARY, OR CONSEQUENTIAL  * DAMAGES (INCLUDING, BUT NOT LIMITED
TO, PROCUREMENT OF SUBSTITUTE GOODS OR  * SERVICES; LOSS OF USE, DATA, OR P
ROFITS; OR BUSINESS INTERRUPTION) HOWEVER  * CAUSED AND ON ANY THEORY OF LI
ABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,  * OR TORT (INCLUDING NEGLI
GENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE  * OF THIS SOFTWARE,
EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.  * *****
***** */

2.  /* Includes -----*/
3.  #include "stm32f4xx_hal.h"
4.  #define TP1 HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_RESET);
5.  HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, GPIO_PIN_RESET)
6.  #define TP2 HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_RESET);
7.  HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, GPIO_PIN_SET)
8.  #define TP3 HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_SET);
9.  HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, GPIO_PIN_RESET)
10. #define NC HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_SET);
11. HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, GPIO_PIN_SET)
12. #define DETECT HAL_GPIO_WritePin(GPIOD, GPIO_PIN_12, GPIO_PIN_SET)
```

```

13. #define UNDETECT HAL_GPIO_WritePin(GPIOD, GPIO_PIN_12, GPIO_PIN_RESET)
14. #define BUZZon HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_SET)
15. #define BUZZoff HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_RESET)
16. #define Tombol1 HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_13)
17. #define Tombol2 HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_15)
18. #define Tombol3 HAL_GPIO_ReadPin(GPIOD, GPIO_PIN_0)
19. UART_HandleTypeDef huart2;
20. void SystemClock_Config(void);
21. static void MX_GPIO_Init(void);
22. static void MX_USART2_UART_Init(void);
23. void recog();
24. uint8_t inRX[20];
25. uint8_t pp[20];
26. uint8_t satu[20];
27. uint8_t dua[20];
28. uint8_t tiga[20];
29. int prt;
30. int one;
31. int two;
32. int three;
33. int status_print;
34. int status_buzzer;
35. int a;
36. int main(void)
37. {
38.     HAL_Init();
39.     pp[0] = 0xAA;
40.     pp[1] = 0x0B;
41.     pp[2] = 0x0D;
42.     pp[3] = 0x00;
43.     pp[4] = 0xFF;
44.     pp[5] = 0x00;
45.     pp[6] = 0x00;
46.     pp[7] = 0x04;
47.     pp[8] = 0x50;
48.     pp[9] = 0x72;
49.     pp[10] = 0x63;
50.     pp[11] = 0x74;
51.     pp[12] = 0x0A;
52.     satu[0] = 0xAA;
53.     satu[1] = 0x0C;
54.     satu[2] = 0x0D;
55.     satu[3] = 0x00;

```

```

56.     satu[4] = 0xFF;
57.     satu[5] = 0x01;
58.     satu[6] = 0x01;
59.     satu[7] = 0x05;
60.     satu[8] = 0x73;
61.     satu[9] = 0x61;
62.     satu[10] = 0x74;
63.     satu[11] = 0x75;
64.     satu[12] = 0x4D;
65.     satu[13] = 0x0A;
66.     dua[0] = 0xAA;
67.     dua[1] = 0x0B;
68.     dua[2] = 0x0D;
69.     dua[3] = 0x00;
70.     dua[4] = 0xFF;
71.     dua[5] = 0x02;
72.     dua[6] = 0x02;
73.     dua[7] = 0x04;
74.     dua[8] = 0x64;
75.     dua[9] = 0x75;
76.     dua[10] = 0x61;
77.     dua[11] = 0x4D;
78.     dua[12] = 0x0A;
79.     tiga[0] = 0xAA;
80.     tiga[1] = 0x0C;
81.     tiga[2] = 0x0D;
82.     tiga[3] = 0x00;
83.     tiga[4] = 0xFF;
84.     tiga[5] = 0x03;
85.     tiga[6] = 0x03;
86.     tiga[7] = 0x05;
87.     tiga[8] = 0x74;
88.     tiga[9] = 0x69;
89.     tiga[10] = 0x67;
90.     tiga[11] = 0x61;
91.     tiga[12] = 0x4D;
92.     tiga[13] = 0x0A;
93.     SystemClock_Config();
94.     MX_GPIO_Init();
95.     MX_USART2_UART_Init();
96.     recog();
97.     while (1) {
98.         NC;

```

```

99.         HAL_UART_Receive( & huart2, inRX, sizeof(inRX), 1000);
100. //Pembacaan Data dan pengelompokkan
101.     prt = 0;
102.     one = 0;
103.     two = 0;
104.     three = 0;
105.     a = 0;
106.     for (int i = 0; i < 13; i++)
107.     {
108.         if (inRX[i] == pp[i])
109.         {
110.             prt = prt + 1;
111.         }
112.         if (inRX[i] == satu[i])
113.         {
114.             one = one + 1;
115.         }
116.         if (inRX[i] == dua[i])
117.         {
118.             two = two + 1;
119.         }
120.         if (inRX[i] == tiga[i])
121.         {
122.             three = three + 1;
123.         }
124.     }
125. // Aktuator Test Print
126.     if (prt > 7 && status_print == 0)
127.     {
128.         status_print = 1;
129.         BUZZon;
130.         HAL_Delay(500);
131.         BUZZoff;
132.         HAL_Delay(500);
133.         BUZZon;
134.         HAL_Delay(500);
135.         BUZZoff;
136.         DETECT;
137.         NC;
138.     }
139.
140.
141.

```

```

142. //Aktuator Receive UART
        else if (one > 10 && status_print == 1)
143.     {
144.         TP1;
145.         status_print = 0;
146.         UNDETECT;
147.         HAL_Delay(8000);
148.         NC;
149.     }
150.     else if (two > 8 && status_print == 1)
151.     {
152.         TP2;
153.         status_print = 0;
154.         UNDETECT;
155.         HAL_Delay(8000);
156.         NC;
157.     }
158.     else if (three > 10 && status_print == 1)
159.     {
160.         TP3;
161.         status_print = 0;
162.         UNDETECT;
163.         HAL_Delay(8000);
164.         NC;
165.     }
166. //Tombol Manual
167.     if (!Tombol1)
168.     {
169.         while (!Tombol1);
170.         BUZZon;
171.         HAL_Delay(3000);
172.         BUZZoff;
173.         TP1;
174.         HAL_Delay(8000);
175.         NC;
176.     }
177.     if (!Tombol2)
178.     {
179.         while (!Tombol2);
180.         BUZZon;
181.         HAL_Delay(3000);
182.         BUZZoff;
183.         TP2;

```

```

184.         HAL_Delay(8000);
185.         NC;
186.     }
187.     if (!Tombol3)
188.     {
189.         while (!Tombol3);
190.         BUZZon;
191.         HAL_Delay(3000);
192.         BUZZoff;
193.         TP3;
194.         HAL_Delay(8000);
195.         NC;
196.     }
197. }
198. }
199. void recog()
200. {
201.     //Clear
202.     uint8_t pData[1];
203.     pData[0] = 0xAA;
204.     HAL_UART_Transmit( & huart2, pData, sizeof(pData), 10);
205.     pData[0] = 0x02;
206.     HAL_UART_Transmit( & huart2, pData, sizeof(pData), 10);
207.     pData[0] = 0x31;
208.     HAL_UART_Transmit( & huart2, pData, sizeof(pData), 10);
209.     pData[0] = 0x0A;
210.     HAL_UART_Transmit( & huart2, pData, sizeof(pData), 10);
211.     HAL_Delay(1000);
212.     //LOAD
213.     pData[0] = 0xAA;
214.     HAL_UART_Transmit( & huart2, pData, sizeof(pData), 10);
215.     pData[0] = 0x06;
216.     HAL_UART_Transmit( & huart2, pData, sizeof(pData), 10);
217.     pData[0] = 0x30;
218.     HAL_UART_Transmit( & huart2, pData, sizeof(pData), 10);
219.     pData[0] = 0x00; //Index TestPrint
220.     HAL_UART_Transmit( & huart2, pData, sizeof(pData), 10);
221.     pData[0] = 0x01; //Index Satu
222.     HAL_UART_Transmit( & huart2, pData, sizeof(pData), 10);
223.     pData[0] = 0x02; //Index Dua
224.     HAL_UART_Transmit( & huart2, pData, sizeof(pData), 10);
225.     pData[0] = 0x03; //Index Tiga
226.     HAL_UART_Transmit( & huart2, pData, sizeof(pData), 10);

```



```

227.     pData[0] = 0x0A;
228.     HAL_UART_Transmit( & huart2, pData, sizeof(pData), 10);
229. }
230. void SystemClock_Config(void) {
231.     RCC_OscInitTypeDef RCC_OscInitStruct;
232.     RCC_ClkInitTypeDef RCC_ClkInitStruct;
233.     __HAL_RCC_PWR_CLK_ENABLE();
234.     __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1);
235.     RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
236.     RCC_OscInitStruct.HSIState = RCC_HSI_ON;
237.     RCC_OscInitStruct.HSICalibrationValue = 16;
238.     RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
239.     RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
240.     RCC_OscInitStruct.PLL.PLLM = 8;
241.     RCC_OscInitStruct.PLL.PLLN = 168;
242.     RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
243.     RCC_OscInitStruct.PLL.PLLQ = 4;
244.     if (HAL_RCC_OscConfig( & RCC_OscInitStruct) != HAL_OK) {
245.         _Error_Handler(__FILE__, __LINE__);
246.     }    /**Initializes the CPU, AHB and APB busses clocks */
247.     RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK | RCC_CLOCKTYPE_SY
        SCLK | RCC_CLOCKTYPE_PCLK1 | RCC_CLOCKTYPE_PCLK2;
248.     RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
249.     RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
250.     RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV4;
251.     RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV2;
252.     if (HAL_RCC_ClockConfig( & RCC_ClkInitStruct, FLASH_LATENCY_5) !=
        HAL_OK) {
253.         _Error_Handler(__FILE__, __LINE__);
254.     }    /**Configure the Systick interrupt time */
255.     HAL_SYSTICK_Config(HAL_RCC_GetHCLKFreq() / 1000);    /**Configure
        the Systick */
256.     HAL_SYSTICK_CLKSourceConfig(SYSTICK_CLKSOURCE_HCLK); /* SysTick_IRQn i
        nterrupt configuration */
257.     HAL_NVIC_SetPriority(SysTick_IRQn, 0, 0);
258. } /* USART2 init function */
259. static void MX_USART2_UART_Init(void) {
260.     huart2.Instance = USART2;
261.     huart2.Init.BaudRate = 9600;
262.     huart2.Init.WordLength = UART_WORDLENGTH_8B;
263.     huart2.Init.StopBits = UART_STOPBITS_1;

```

```

264.     huart2.Init.Parity = UART_PARITY_NONE;
265.     huart2.Init.Mode = UART_MODE_TX_RX;
266.     huart2.Init.HwFlowCtl = UART_HWCONTROL_NONE;
267.     huart2.Init.OverSampling = UART_OVERSAMPLING_16;
268.     if (HAL_UART_Init( & huart2) != HAL_OK) {
269.         _Error_Handler(__FILE__, __LINE__);
270.     }
271. } /** Configure pins as          * Analog          * Input          * Output
    tput          * EVENT_OUT          * EXTI*/
272. static void MX_GPIO_Init(void) {
273.     GPIO_InitTypeDef GPIO_InitStruct; /* GPIO Ports Clock Enable */

274.     __HAL_RCC_GPIOA_CLK_ENABLE();
275.     __HAL_RCC_GPIOD_CLK_ENABLE();
276.     GPIO_InitStruct.Pin = GPIO_PIN_13 | GPIO_PIN_15;
277.     GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
278.     GPIO_InitStruct.Pull = GPIO_PULLUP;
279.     GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
280.     HAL_GPIO_Init(GPIOA, & GPIO_InitStruct);
281.     GPIO_InitStruct.Pin = GPIO_PIN_0;
282.     GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
283.     GPIO_InitStruct.Pull = GPIO_PULLUP;
284.     GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
285.     HAL_GPIO_Init(GPIOD, & GPIO_InitStruct); /*Configure GPIO pin Output
    t Level */
286.     HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13 | GPIO_PIN_14 | GPIO_PIN_15,
        GPIO_PIN_RESET); /*Configure GPIO pins : PD13 PD14 PD15 */
287.     GPIO_InitStruct.Pin = GPIO_PIN_13 | GPIO_PIN_14 | GPIO_PIN_15;

288.     GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
289.     GPIO_InitStruct.Pull = GPIO_NOPULL;
290.     GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
291.     HAL_GPIO_Init(GPIOD, & GPIO_InitStruct);
292. } /* USER CODE BEGIN 4 */ /* USER CODE END 4 */ /** * @brief Th
    is function is executed in case of error occurrence. * @param file: The f
    ile name as string. * @param line: The line in file as a number. * @retv
    al None */
293. void _Error_Handler(char * file, int line) { /* USER CODE BEGIN Erro
    r_Handler_Debug */ /* User can add his own implementation to report the H
    AL error return state */
294.     while (1) {} /* USER CODE END Error_Handler_Debug */
295. }

```

```

296. #ifndef USE_FULL_ASSERT /** * @brief Reports the name of the source fil
    e and the source line number *          where the assert_param error has oc
    curred. * @param file: pointer to the source file name * @param line: a
    ssert_param error line source number * @retval None */ void assert_fa
    iled(uint8_t * file, uint32_t line) { /* USER CODE BEGIN 6 */
    /* User can add his own implementation to report the file name and line num
    ber, tex: printf("Wrong parameters value: file %s on line %d\r\n", file
    , line) */ /* USER CODE END 6 */ }
297. #endif /* USE_FULL_ASSERT */ /** * @} */ /** * @} */ /* *****
    ***** (C) COPYRIGHT STMicroelectronics *****END OF FILE****/

```

-----*Halaman ini sengaja dikosongkan*-----

## BIODATA PENULIS



Penulis adalah seorang pria yang dilahir di Jakarta 26 September 1996. Dilahirkan dari seorang Ibu bernama Yohan Avila Nawang Windarti dan seorang Ayah bernama Pius Benny Benjamin sebagai anak pertama laki-laki satu-satunya, penulis memiliki dua orang adik perempuan bernama Archangela Reza Serafine Anindita Putri dan Michele Reza Zefanya Elena Putri. Memulai pendidikan di SD Marsudirini Bekasi lalu SMP Cahaya Harapan lalu SMA Marsudirini Bekasi. Semenjak SMP berminat pada Elektronika dan bercita-cita menjadi

Sarjana Teknik Elektro. Karena itu melanjutkan pendidikan di ITS Surabaya pada tahun 2014. Selama menjalani masa perkuliahan di ITS, penulis mengikuti beberapa organisasi dan kepanitian salah satunya di Himatektro dan Pemandu ITS, selain itu juga menjadi asisten praktikum Rangkaian Listrik dan Rangkaian Elektronika.

Email: [nicolas.rupiah@gmail.com](mailto:nicolas.rupiah@gmail.com)/[nicolas.rezadhi14@mhs.ee.its.ac.id](mailto:nicolas.rezadhi14@mhs.ee.its.ac.id)

